

ESTUDIO DE LA VIABILIDAD DE UN ASISTENTE
ACCESIBLE PARA LA GESTIÓN DE ESPACIOS
PÚBLICOS

VIABILITY STUDY OF AN ACCESSIBLE ASISTANT FOR
THE MANAGEMENT OF PUBLIC SPACES



TRABAJO FIN DE MÁSTER
CURSO 2019-2020

AUTOR
ADRIÁN CONDE LÓPEZ

DIRECTOR
MARÍA GUIJARRO MATA-GARCÍA / JOAQUÍN RECAS PIORNO

MÁSTER EN INTERNET DE LAS COSAS
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

ESTUDIO DE LA VIABILIDAD DE UN ASISTENTE ACCESIBLE PARA LA GESTIÓN DE ESPACIOS PÚBLICOS

VIABILITY STUDY OF AN ACCESIBLE ASISTANT FOR THE MANAGEMENT OF PUBLIC SPACES

TRABAJO DE FIN DE MÁSTER EN INTERNET DE LAS COSAS
DEPARTAMENTO DE ARQUITECTURA DE ORDENADORES Y AUTOMÁTICA

AUTOR
ADRIÁN CONDE LOPEZ

DIRECTOR
MARÍA GUIJARRO MATA-GARCÍA / JOAQUÍN RECAS PIORNO

CONVOCATORIA: SEPTIEMBRE 2020
CALIFICACIÓN: 6.5

MÁSTER EN INTERNET DE LAS COSAS
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

23 DE SEPTIEMBRE DE 2020

DEDICATORIA

A todos aquellos que me han animado
durante estos meses a seguir adelante.

AGRADECIMIENTOS

A mis profesores, que me han enseñado muchas cosas que desconocía este año de diversos temas, ampliando o mejorando mis conocimientos. Con lo que he podido realizar un mejor estudio en este trabajo.

A mis tutores María y Joaquín, que me han ayudado todo lo que han podido con la elección del tema y con el desarrollo del estudio.

Y por supuesto a mis compañeros del máster, que siempre que lo he necesitado han estado ahí.

RESUMEN

Estudio de la Viabilidad de un Asistente Accesible para la Gestión de Espacios Públicos

Es una clara evidencia que el mundo en el que vivimos no ha parado de cambiar e innovar cada día que ha pasado. El propósito suele ser el tópico de siempre, mejorar la vida de los seres humanos. Si, por ejemplo, nos ponemos en la situación de que un grupo de personas quiere entrar en un lugar, o espera ser atendido por alguien en concreto, los seres humanos hacemos lo que se denomina comúnmente como una cola. Esta la realizamos de "a uno", sin embargo, desde hace algún tiempo hacemos uso de los tiques. Cada persona tiene un tique y cada tique tiene asignado un número que indica su turno, y los turnos van apareciendo en una pantalla. Si observamos esta realidad con perspectiva, ¿qué pasa con aquellas personas que sufren de algún tipo de discapacidad visual? No pueden saber cuando es su turno sin tener que hacer uso de la humildad humana. En este trabajo se analiza una posible solución enfocada principalmente en estas personas, con el fin de seguir mejorando la vida de todos los seres humanos.

Palabras clave

Accesibilidad, tique, OCR,

ABSTRACT

Viability Study of an Accesible Asistant for the Management of Public Spaces

It is clear the world where we live has not stopped changing and innovating every day that has passed. Purpose usually is the usual topic, to improve the lives of human beings, making it easier. If we put ourselves, for example, in a situation where a group of people wants to enter a place, or expects to be attended by someone in particular, we do what is known as queue. We usually do this like "one at a time", however, for some time now, we use tickets. Each person has a ticket, which is assigned with a number that indicates their turn and turns appear on a screen. Now we are going to look this reality with perspective, what is going on to those people who has some type of visual disability? They cannot know when it is their turn without having to use human humility. In this project, a possible solution is being studied focusing on these people, in order to continue improving the lives of all human beings.

Keywords

Accessibility, ticket, OCR,

ÍNDICE DE CONTENIDOS

Dedicatoria	III
Agradecimientos.....	V
Resumen.....	VII
Abstract	IX
Índice de contenidos	X
Índice de figuras.....	XIV
Índice de tablas	XVII
Capítulo 1 - Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Plan de trabajo.....	3
1.4 Estructura de la memoria.....	4
Capítulo 2 - Estado de la cuestión	5
2.1 TalkBack.....	5
2.2 VoiceOver	5
2.3 Be My Eyes	5
2.4 Seeing AI	6
2.5 Change.org	7
Capítulo 3 - Presentación de los Recursos	9
3.1 Raspberry Pi 2 Model B	9
3.2 Adaptador Bluetooth SMCBT-EDR.....	10
3.3 Android STUDIO	10
3.4 Teléfono Móvil Just5 M503.....	11

Capítulo 4 - Red de Comunicación	13
4.1 WiFi	13
4.1.1 Ventajas.....	14
4.1.2 Desventajas.....	14
4.2 Bluetooth	14
4.2.1 Versiones de Bluetooth	16
4.2.2 Ventajas.....	17
4.2.3 Desventajas.....	19
4.3 ZigBee	19
4.3.1 Protocolo	20
4.3.2 Topologías	20
4.3.3 Tipos de Nodos	21
4.3.4 Ventajas.....	21
4.3.5 Desventajas.....	21
4.4 Z-Wave.....	22
4.4.1 Ventajas.....	23
4.4.2 Desventajas.....	24
4.5 Conclusiones Red de Comunicación	24
Capítulo 5 - Reconocimiento Óptico de Caracteres.....	27
5.1 Introducción	27
5.2 Funcionamiento	27
5.3 Aprendizaje Profundo.....	28
5.4 Tesseract - OCR	28
5.4.1 Funcionamiento	28
5.4.2 Búsqueda de línea	30

5.4.3 Reconocimiento de palabras	32
5.4.4 Clasificador de caracteres estático	34
5.4.5 Análisis Lingüístico.....	36
5.4.6 Clasificador adaptativo	37
5.5 Conclusiones OCR	38
Capítulo 6 - Aplicación	39
6.1 Interfaz (Front End)	39
6.2 Back End	43
6.2.1 Activity_main.....	44
6.2.2 MainActivity	44
6.2.3 Notifications	44
6.3 Accesibilidad de la aplicación	45
Capítulo 7 - Raspberry Pi	47
7.1 Reconocimiento óptico de caracteres	47
7.2 Recepción de las imágenes por bluetooth.....	49
7.3 Envío de notificaciones a la aplicación.....	51
Capítulo 8 - Pruebas Realizadas.....	53
8.1 Introducción	53
8.2 Parámetros Tesseract.....	53
8.3 Pruebas.....	53
Capítulo 9 - Asignaturas del Máster en Internet de las Cosas.....	63
9.1 Arquitectura del Nodo IoT.....	63
9.2 Diseño de Infraestructura inteligente para el IoT	63
9.3 Tratamiento de Datos Masivos / Inteligencia Artificial Aplicada al Internet de las Cosas	63

9.4 Redes, Protocolos e Interfaces (I & II)	63
9.5 Seguridad y Legalidad	63
Capítulo 10 - Conclusiones y trabajo futuro	65
10.1 Introducción	65
10.2 Metas alcanzadas	65
10.3 Problemas encontrados	65
10.4 Trabajo futuro	66
Capítulo 11 - Introduction (English)	67
11.1 Motivation	67
11.2 Objectives	67
11.3 Work plan	69
11.4 Memory structure	70
Capítulo 12 - Conclusions and future work (English)	71
12.1 Introduction.....	71
12.2 Goals achieved.....	71
12.3 Problems encountered.....	71
12.4 Future work.....	72
Bibliografía	73

ÍNDICE DE FIGURAS

Figura 2-1. Logotipo de "Be My Eyes". Fuente: Be My Eyes.....	6
Figura 3-1. Raspberry Pi 2 Model B. Fuente: Raspberryshop	9
Figura 3-2. Adaptador Bluetooth SMCB-EDR. Fuente: CNET	10
Figura 3-3. Logo Android STUDIO. Fuente: Android Studio	10
Figura 3-4. Teléfono Móvil Just5 M503. Fuente: AMAZON.....	11
Figura 4-1. Popular icono que representa el WiFi. Fuente: Freepik	13
Figura 4-2. Logotipo del Bluetooth. Fuente: 100marcas.....	15
Figura 4-3. Ley Inversa del Cuadrado. Fuente: Repetidores Móviles.....	18
Figura 4-4. Logotipo de ZigBee. Fuente: Developer Bosch IoT Suite	19
Figura 4-5. Logotipo Z-Wave. Fuente: ROBOTechnics	22
Figura 4-6. Diagrama de interconexión de los equipos	25
Figura 4-7. Diagrama de procesos.....	25
Figura 5-1. Ejemplo de línea de base ajustada a curva. Fuente: Optical Character Recognition.....	31
Figura 5-2. Dificultades de espaciado de texto. Fuente: Optical Character Recognition	32
Figura 5-3. Candidatos de puntos de corte y corte. Fuente: The Extraction and Recognition of Text from Multimedia Document Images	32
Figura 5-4. Palabra fácilmente reconocible. Fuente: The Extraction and Recognition of Text from Multimedia Document Images	34
Figura 5-5. (a) "h" prístina, (b) "h" rota, (c) característica emparejas con prototipos. Fuente: Omnidocument Technologies	34
Figura 5-6. Letras en forma normalizada de línea de base/altura x y forma normalizada de momento. Fuente: Automatic Prototype Extraction for Adaptive OCR.....	38

Figura 6-1. imagen de desarrollo de la aplicación.....	39
Figura 6-2. Foto de la aplicación (Just5 M503 y Google Pixel Simulado)	40
Figura 6-3. Al presionar el botón, nos abre la aplicación de la cámara (Just5 M503 y Google Pixel Simulado)	41
Figura 6-4. Una vez sacada la foto, debemos aceptarla (Just5 M503 y Google Pixel Simulado).....	41
Figura 6-5. Vista previa de la foto (Just5 M503 y Google Pixel Simulado)	42
Figura 6-6. Pantalla de notificaciones (Just5 M503 y Google Pixel Simulado)	43
Figura 7-1. Versión sistema operativo Raspberry	47
Figura 7-2. Instalación Tesseract en la Raspberry Pi.....	48
Figura 7-3. Lista de idiomas instalados de Tesseract.....	48
Figura 7-4. Pruebas ejecución de Tesseract	49
Figura 7-5. Instalación Blueman.....	49
Figura 7-6. Emparejamiento por bluetooth (1/3)	50
Figura 7-7. Emparejamiento por bluetooth (2/3)	50
Figura 7-8. Emparejamiento por bluetooth (3/3)	51
Figura 8-1. Archivo de entrada 1. Fuente: EnfermeriaDeCiudadReal.....	54
Figura 8-2. Archivo de entrada 2. Fuente: Gandía Departament de Salut.....	54
Figura 8-3. Archivo de entrada 3. Fuente: Sanidad Castilla La Mancha	55
Figura 8-4. Archivo de entrada 4. Fuente: Change.org	56
Figura 8-5. Archivo de salida 4	56
Figura 8-6. Archivo de entrada 5. Fuente: 20minutos.....	57
Figura 8-7. Archivo de salida 5	57
Figura 8-8. Archivo de entrada 6. Fuente: GESTORN	58
Figura 8-9. Archivo de salida 6	58

Figura 8-10. Archivo de entrada 7. Fuente: GESTORN	59
Figura 8-11. Archivo de salida 7	59
Figura 8-12. Archivo de entrada 8. Fuente: Hospital de l'Espirit Sant	60
Figura 8-13. Archivo de salida 8	60

ÍNDICE DE TABLAS

Tabla 1-1. Organización del trabajo	3
Tabla 4-1. Velocidad WiFi por protocolo.....	14
Tabla 4-2. Alcance Bluetooth	15
Tabla 4-3. Protocolos de ZigBee	20
Tabla 9-1. Esquema visual relación TFM-Asignaturas.....	64
Tabla 11-1. Work organization	69

Capítulo 1 - Introducción

1.1 Motivación

Gracias a mis tutores llegamos al tema escogido que me parece muy interesante y que podría ayudar a muchas personas. Con lo que la motivación es doble, por un lado, la de conseguir un tema para realizar dicho trabajo final de máster y por otro el realizar una tarea que podría ser de utilidad para personas que realmente lo necesitan. En un principio, habíamos comentado la posibilidad de realizar un asistente de compra. En concreto, realizar el diseño e implementación de un prototipo de asistente para la compra en supermercado para personas ciegas. El escenario sería el siguiente, las estanterías del supermercado estarían balizadas (con dispositivos a elección), de forma que la persona invidente reciba información sonora de la localización de los productos de su lista de la compra. Con lo que, una vez en la zona de interés, se reconocería el producto exacto mediante visión artificial usando, por ejemplo, la cámara del móvil.

Analizando esta propuesta, observamos que la complejidad de este proyecto era bastante considerable, ya que habría que hacer un escaneado de todos los productos, la cámara tendría que ponerla la persona de forma correcta para analizar el producto etcétera. Además, también existe la compra por internet, con lo que le dimos una vuelta al tema del trabajo.

1.2 Objetivos

Esto nos hizo llegar el tema de investigación, un asistente accesible para la gestión de espacios públicos. La idea es sencilla, cualquier edificio público que cuenta con algún tipo de gestión al ciudadano, lo cual conlleva el realizar una cola, cuenta con el típico stand en el que, al llegar, eliges el departamento al que necesitar ir, sale un tique con nuestro número y automáticamente entras en una "cola virtual", donde cuando sea nuestro turno el número que se nos ha asignado aparecerá en una pantalla.

Pues bien, ¿qué pasaría si una persona ciega necesita realizar este tipo de trámites o simplemente ir al médico (ya que se está comenzando a implantar cada vez en más centros de salud) teniendo que pasar por este sistema virtual?

Así es como comenzamos a diseñar este trabajo. La persona ciega llegaría al edificio, se colocaría delante del stand y con una aplicación de su móvil sacaría una foto al tique. Esa foto se enviaría, en este caso, a una Raspberry pi 2 Model b [1], donde se realizaría un reconocimiento óptico de caracteres para conseguir ese turno en un lenguaje que el sistema pudiera comprender. Ahora vamos a ver como sabría la persona que su número ha salido en pantalla y donde tendría que ir. En la segunda parte del diseño nos encontramos con unas cámaras apuntando a las pantallas donde van saliendo los turnos y las salas. Supongamos que tenemos tres plantas con una posible sala por planta. Tendríamos entonces tres cámaras apuntando a las respectivas pantallas. Dichas cámaras estarían conectadas a nuestra Raspberry. Con lo que, avanzando en la situación, la persona ha llegado al centro, ha sacado una fotografía a su tique y esa fotografía ha llegado a la Raspberry. Lo primero, tras realizar el reconocimiento óptico de caracteres, sería saber a que planta ha de ir la persona. Esto lo sacaríamos gracias a lo obtenido tras el OCR [2], y lo enviaríamos a la aplicación que se lo diría a la persona. Imaginémonos que sale la segunda planta, la persona subiría a la primera planta, y la aplicación le diría que todavía le falta una planta por subir, en el momento que llegase a la segunda planta, la aplicación le diría que ha llegado a la planta correcta y tendría que esperar.

Con respecto a la cámara de la segunda planta enfocada a esa pantalla de turnos. La cámara iría sacando "capturas" cada cierto tiempo o cada vez que varían las posiciones y las enviaría a nuestra Raspberry. En la Raspberry, como con el tique del stand del turno, se enviarían las imágenes por el reconocimiento óptico de caracteres y en el momento que coincida el turno que sale en la pantalla con el del tique de nuestro turno, se notificaría mediante la aplicación al usuario de que su turno ha llegado.

Es un sistema no intrusivo, lo que quiere decir que podría existir un mejor desarrollo en el caso de poder estar dentro de la red que exista en el centro. Pero se ha desarrollado de tal forma que no interfiera en lo que ya exista.

1.3 Plan de trabajo

Analizando el prototipo en el que se enmarca este trabajo, lo he dividido en tres ramas de estudio que a mi juicio serían las principales:

- a) Para empezar la red de comunicación, como conectar los dispositivos que entran en juego.
- b) El reconocimiento óptico de caracteres, ya que se necesita obtener el turno de la persona y compararla con los que aparezcan en pantalla.
- c) Aplicación, la vía mediante la que el usuario interaccionará con todo el sistema.

En la siguiente tabla (tabla 1-1) se puede ver un diagrama de Gantt de cómo se ha organizado dicho trabajo teniendo en cuenta las tareas y las semanas.

TAREA	S.1	S.2	S.3	S.4	S.5	S.6	S.7	S.8	S.9	S.10	S.11	S.12	S.13	S.14	S.15
Red de Comunicación															
OCR															
Aplicación															
Memoria															

Tabla 1-1. Organización del trabajo

1.4 Estructura de la memoria

En este apartado veremos de una forma general cada uno de los apartados del presente documento. La estructura es la siguiente:

- **Capítulo 1: Introducción.** Introducción del proyecto en el que entra la motivación, junto con los objetivos y el plan de trabajo de este.
- **Capítulo 2: Estado de la cuestión.** Se explican proyectos, que en cierta manera tienen que ver con el estudiado en el presente documento.
- **Capítulo 3: Presentación de los recursos.** Breve presentación de los dispositivos usados para las pruebas del proyecto.
- **Capítulo 4: Red de comunicación.** Análisis de las diferentes topologías de red con el fin de llegar a la que más convendría.
- **Capítulo 5: Reconocimiento óptico de caracteres.** Estudio de los diferentes softwares para la realización del reconocimiento óptico de caracteres (OCR).
- **Capítulo 6: Aplicación.** Se muestra la prueba de aplicación desarrollada con la que el usuario final interactuaría.
- **Capítulo 7: Raspberry Pi.** Análisis de todo lo desarrollado en uno de los dispositivos.
- **Capítulo 8: Pruebas realizadas:** Test realizados con el programa de reconocimiento óptico de caracteres Tesseract.
- **Capítulo 9: Asignaturas del máster en internet de las cosas.** Breve resumen de la ayuda que ha sido cursar las diferentes asignaturas del máster para el desarrollo de este proyecto.
- **Capítulo 10: Conclusiones y trabajo futuro.** Conclusiones del presente proyecto y trabajo futuro.
- **Capítulo 11: Introduction (English).** Capítulo 1 en inglés.
- **Capítulo 12: Conclusions and future work (English).** Capítulo 10 en inglés.

Capítulo 2 - Estado de la cuestión

Tras una exhaustiva revisión del estado de la cuestión, se encuentran cinco proyectos de relevancia que de alguna manera están relacionados con el presente estudio.

2.1 TalkBack

Algo tan simple como puede ser realizar una llamada, o enviar un mensaje de texto, para una persona que sufra algún tipo de discapacidad visual puede ser toda una odisea. “TalkBack” es el lector de pantalla de Google incluido en los dispositivos Android. Este servicio ofrece al usuario mensajes de voz que permite usar el dispositivo sin que tener que mirar la pantalla. [3] Hasta hace no mucho tiempo, TalkBack era una aplicación independiente que se podía descargar a través de la tienda de Android Google Play Store para cualquier dispositivo. Hoy en día, el servicio llega preinstalado en la mayoría de los dispositivos como parte del paquete “Android Accessibility Suite” [4].

2.2 VoiceOver

Sin embargo, no solo Android apuesta por la inclusión de las personas invidentes. VoiceOver es un lector de pantalla con el que el usuario puede disfrutar de los dispositivos de Apple, aunque no los puedan ver. Con este asistente activado, basta con pulsar tres veces un botón para abrirle desde cualquier lugar de iOS. Con ello se consigue escuchar una descripción de todo lo que hay en la pantalla, desde el nivel de batería hasta quién está llamando o el nombre de la aplicación sobre la que el usuario tiene puesto el dedo. [5]

2.3 Be My Eyes

“Be My Eyes” se creó para ayudar a personas ciegas o con problemas de visión. La aplicación está compuesta por una comunidad global de ciegos y personas con problemas de visión y voluntarios sin discapacidad visual. Be My Eyes toma el poder de la tecnología y la conexión humana para llevar vista a las personas con pérdida de visión. A través de una video llamada en directo, los voluntarios asisten a ciegos y

personas con déficit de visión prestándoles ayuda visual para tareas como el reconocimiento de colores, comprobar si las luces están encendidas o preparar la cena. La aplicación es de uso gratuito y está disponible para iOS y Android. [6]



Figura 2-1. Logotipo de "Be My Eyes". Fuente: Be My Eyes

La aplicación es capaz de ofrecer una gran asistencia a personas que realmente lo necesiten, pero podemos ver que el usuario dependería de que hubiera una persona al otro lado. En la aplicación que estoy planteando en este trabajo no existiría este problema ya que sería totalmente autónoma de que hubiera una persona “al otro lado”.

2.4 Seeing AI

Aplicación desarrollada por Microsoft y especialmente diseñada para personas ciegas o con dificultades visuales, es capaz de describir lo que el usuario tiene delante cuando lo enfoca con la cámara. Incluye funcionalidades como la lectura de textos cortos, escaneo de códigos de barras, reconocimiento facial, determinar el valor de billetes y divisas y reconocimiento del entorno entre otros. [7]

Es una aplicación que, consultando testimonios de usuarios, ayuda a estas personas en el día a día. Sin embargo, también tiene cosas a mejorar. Por ejemplo, la importancia de las descripciones “a veces dice barbaridades”. Al final la aplicación realiza una descripción muy crítica y básica, que sirve al usuario para “hacerse una idea de lo que hay”. [8]

2.5 Change.org

Change.org es una página desde la cual cualquier persona puede iniciar una petición de forma gratuita [9]. Navegando entre las diferentes peticiones, podemos ver una que inició Ismael Calle. Ismael es una persona ciega que acude al Hospital del Henares de Coslada (Madrid). En la petición que ha iniciado hace ver el problema que vemos en este trabajo, ya que en algunos hospitales (como el suyo), han instalado unas nuevas pantallas para organizar a los pacientes y distribuirlos en turnos y sales. Sin embargo, estas pantallas no van acompañadas de ningún aviso sonoro, con lo que desde que se han puesto en marcha, las personas ciegas no saben cuando les toca ni a que sale deben pasar. En la petición, Ismael no critica el nuevo sistema, que ha demostrado ser más eficiente, sino que exista algún tipo de complemento para que las personas ciegas también puedan disfrutar de sus ventajas. En este trabajo se propone una solución para todas aquellas personas que se sienten como Ismael [10].

Capítulo 3 - Presentación de los Recursos

A continuación, paso a describir los elementos y programas que he usado para realizar este estudio.

3.1 Raspberry Pi 2 Model B

Esta Raspberry es la renovación de la antigua Raspberry Pi 1 Model B + y cuenta con un procesador broadcom ARM Cortex-A7 de cuatro núcleos y 1 GB de RAM. [1]

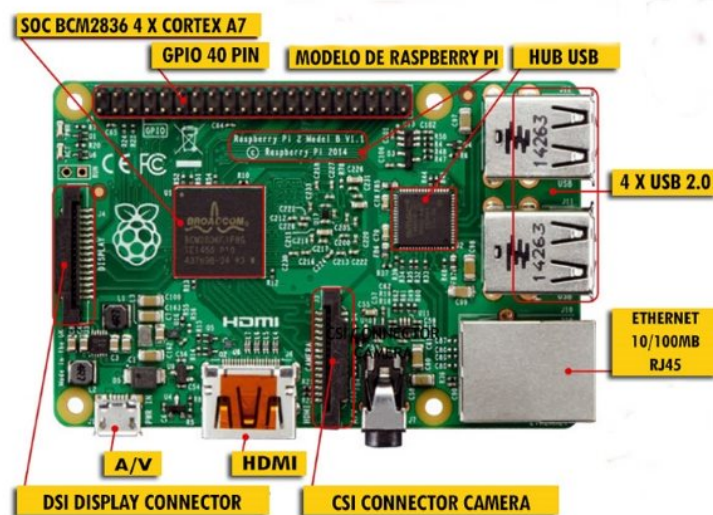


Figura 3-1. Raspberry Pi 2 Model B. Fuente: Raspberryshop

. Además, cuenta con cuatro puertos USB 2.0, un conector GPIO [11] de 40 contactos y HDMI entre otros. Su consumo es de 800 mA (4.0 W) [1]

Aparte de las especificaciones técnicas que son sobradamente suficientes para la realización de este estudio, es una placa muy utilizada. Con lo que existe una gran comunidad de desarrolladores con comunidades en las que comparten código abierto, lo que siempre es beneficioso para el desarrollo de ideas [12].

3.2 Adaptador Bluetooth SMCBT-EDR

Debido a que el modelo de Raspberry utilizado no tiene bluetooth incorporado (ya que se añadió en la siguiente versión), se ha usado el módulo bluetooth SMCBT-EDR para ampliar las interfaces de comunicación de la Raspberry. Integrando un puerto de comunicaciones tan extendido como es el bluetooth, que será imprescindible en las tareas posteriores de trasiego de información entre el usuario y los sistemas de gestión de colas existentes. [13]



Figura 3-2. Adaptador Bluetooth SMCB-EDR. Fuente: CNET

3.3 Android STUDIO

Android STUDIO es el entorno de desarrollo oficial para crear aplicaciones de la plataforma *ANDROID*. Anunciado por Google el 16 de mayo de 2013 y publicada la primera versión estable en diciembre de 2014, nació como una alternativa al entorno *Eclipse* (el cual era el mas usado hasta la fecha). [14]



Figura 3-3. Logo Android STUDIO. Fuente: Android Studio

3.4 Teléfono Móvil Just5 M503

A la hora de probar la aplicación con *Android STUDIO* [14], además de usar el simulador que viene integrado en el programa, he usado el teléfono móvil que también tenía *Just5 M503*. Cuenta con una pantalla de 5 pulgadas que alcanza una resolución HD. Procesador *MediaTek MT6580* de cuatro núcleos que alcanza una velocidad de reloj de hasta 1.3HGz, junto con 1GB de memoria RAM y 8 GB de almacenamiento interno. [15]



Figura 3-4. Teléfono Móvil Just5 M503. Fuente: AMAZON

Capítulo 4 - Red de Comunicación

En este apartado se realizará una prueba de concepto sobre la red de comunicación que, desde el punto de vista del autor, funcionaría mejor para el sistema. Recopilando las comunicaciones que habría que estudiar, se encuentra, por un lado, la existente entre la Raspberry y el móvil y, por otro lado, las cámaras y la Raspberry. También podríamos tener en cuenta para versiones futuras o mejoras que el móvil se comunicase de forma directa con las cámaras. Analizaré las conocidas WiFi y Bluetooth además de ZigBee y Z-Wave.

4.1 WiFi

WiFi (del inglés "*Wireless Fidelity*", "*fideliadad inalámbrica*"), es una tecnología de transmisión de datos de forma inalámbrica basada en el estándar 802.11 [16]. Dicha tecnología se basa en ondas de radio, como funciona la propia radio, la telefonía móvil o la televisión. Difieren, sin embargo, en las frecuencias usadas, en este caso 2,4GHz hasta el estándar 802.11 n y 5GHz en 802.11 ac. Actualmente, en el ámbito que más conocemos esta tecnología, que es para internet, se usan ambas frecuencias. [17]



Figura 4-1. Popular icono que representa el WiFi. Fuente: Freepik

En una conexión mediante WiFi tenemos un adaptador inalámbrico en un dispositivo que su función es "traducir" los datos en forma de señal de radio y, ayudado por una antena, los transmite de forma inalámbrica. Seguidamente nos encontramos con un router, que es el encargado de recibir la señal y decodificarla. A continuación, una vez completado esta conexión, y mediante una conexión física por cable, la información viaja a través de Ethernet, hasta otros servidores. Si, por el contrario, somos

nosotros los que queremos recibir información desde internet, el proceso es justamente el inverso.

Si analizamos las velocidades teóricas atendiendo al protocolo [18], nos encontramos con las siguientes:

Protocolo	Velocidad Máxima Teórica
802.11a	54 Mbps
802.11b	11 Mbps
802.11g	54 Mbps
802.11n	300 Mbps
802.11ac	433 Mbps

Tabla 4-1. Velocidad WiFi por protocolo

4.1.1 Ventajas

Una de las principales ventajas que nos podríamos encontrar con esta tecnología es el alto grado que tiene de aceptación. Con lo que raro sería el dispositivo que usáramos que no hiciera uso de esta. Asimismo, supone también una ventaja nada desdeñable la ausencia de cables a la hora de conectar los dispositivos.

4.1.2 Desventajas

Podrían llegar a darse fallos en la comunicación por saturación de la red. El alcance es limitado para la recepción de la señal, lo que conllevaría a usar múltiples repetidores para lograr un sistema con un alcance mayor. Sin olvidar la “facilidad” de hackeo que tiene esta tecnología. [19]

4.2 Bluetooth

El término Bluetooth define una tecnología de red que fue desarrollada por un consorcio de Ericsson, IBM, Toshiba y Nokia. Invento sueco cuyo símbolo es una ruina vikinga. El grupo de trabajo IEEE 802.15.1 del “*Institute of Electrical and Electronics Engineers*” de estados unidos formalizó el protocolo como estándar industrial para conexiones inalámbricas. Gracias a esta tecnología, podemos transferir voz y datos

punto a punto sin conexión o incluso conectar dos dispositivos digitales diferentes con una radiofrecuencia segura en la banda ISM de los 2,4 GHz [20]. El principal objetivo de dicha tecnología no es otro que el de reemplazar las conexiones por cable.

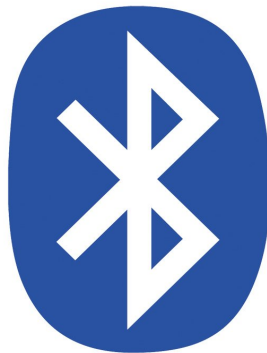


Figura 4-2. Logotipo del Bluetooth. Fuente: 100marcas

Uno de los grandes logros que ha conseguido esta tecnología desde sus inicios es el de no interferir en otros sistemas inalámbricos que operan por radiofrecuencia. Para conseguirlo, el Bluetooth envía señales débiles de menos de 1 milivatio. Con ello, los datos viajan de manera eficaz y sin ser interferidos por otras señales.

Si nos enfocamos en el alcance de esta tecnología, de momento podemos distinguir cuatro tipos:

Tipo	Máxima potencia de transmisión	Máximo alcance (interiores)	Máximo alcance (exteriores)
Tipo 1	100 mW	100 m	200 m
Tipo 2	2,5 mW	10 m	50 m
Tipo 3	1 mW	1 m	10 m
Tipo 4	0,5 mW	0,5 m	5 m

Tabla 4-2. Alcance Bluetooth

Como podemos comprobar y es algo coherente, a medida que disminuye la potencia de transmisión, menor es el alcance.

4.2.1 Versiones de Bluetooth

A continuación, vamos a repasar las distintas versiones que existen [21].

- **Bluetooth 1.0.** Lanzada en 1999, fue la primera versión lanzada de esta tecnología. No obstante, al ser la primera, tuvo muchos problemas en cuanto a conectividad y seguridad. Actualmente se encuentra en desuso.
- **Bluetooth 1.1.** En 2002 llegó la primera actualización del bluetooth 1.0. Llegó como una versión madura y comercial con una tasa de transmisión que rondaba los 721 kbps.
- **Bluetooth 1.2.** La segunda actualización del Bluetooth 1.0 llegó en 2003. Consiguieron reducir las interferencias que existían todavía en la versión 1.1 para seguir puliendo la tecnología.
- **Bluetooth 2.0.** En 2004 aterrizó la segunda gran versión del Bluetooth 1.0, aunque seguía siendo compatible con la 1.2. Su característica principal fue la introducción del a “*Enhanced Data Rate*” (EDR o tasa mejorada de datos). Que permitía ofrecer una tasa de transmisión de algo más de 2 Mb/s.
- **Bluetooth 2.1.** En 2007 llegó la siguiente versión, manteniendo la misma tasa de transmisión que su antecesora. No obstante, introdujo una característica significativa que cambió el Bluetooth para siempre. Y es la posibilidad de que un terminal pudiese agrega a otro, y conectarse automáticamente sin necesidad de un código.
- **Bluetooth 3.0.** En 2009 llegó una nueva versión, que trajo el término HS de High Speed (alta velocidad). Conllevó una gran mejora en la tasa de transmisión, que podía llegar a alcanzar los 24 Mbps.
- **Bluetooth 4.0.** Y llegó una de las versiones mas importantes del estándar, ya que fue el inicio del combate contra el consumo excesivo de batería de la tecnología mediante Bluetooth Low Energy. Esta versión vio la luz en 2010, y manteniendo la tasa de transferencia de 24 Mb/s. Sin embargo, gracias a su menor consumo, comenzó a utilizarse en dispositivos más pequeños y menos potentes.

- **Bluetooth 4.1.** En 2013 llegó una nueva versión, caracterizada por el enfoque hacia nuestro mundo, el IoT (Internet de las Cosas), ya que permitía la conexión entre dispositivos pequeños sin intermediarios.
- **Bluetooth 4.2.** La siguiente actualización llegó en 2014, implementando el protocolo IPv6 para permitir la conexión directa a través de internet.
- **Bluetooth 5.0.** Hacia la mitad del 2016 llegó una nueva revolución. Se doblaba la tasa de transferencia, cuadruplicaba el alcance, continuaba con un bajo consumo teniendo en mente el IoT y permitía una mayor cantidad de datos en cada mensaje transferido. Si hablamos de números, la tasa de transferencia era de hasta 50 Mb/s y el alcance rondaba los 240 m.
- **Bluetooth 5.1.** En 2019 llegó una nueva versión, que se enfocaba en torno a la localización. Permitía que los dispositivos pudieran saber la ubicación de otros dispositivos a los que estén conectados con un margen de centímetros. Incluso podrían identificar la dirección de donde proviene una señal que está buscando.
- **Bluetooth LE Audio.** A principios de este año se publicó un nuevo estándar. Se centró en la mejora de la calidad de sonido y la eficiencia energética en todos los dispositivos que se conectan para funciones relacionadas con el audio. Usando un nuevo códec, el "*Low Complexity Communication Codec*" (LC3), que puede comprimir y descomprimir de forma más eficiente los datos que se transmiten. Además, permiten la transferencia de audio a varios dispositivos a la vez.

4.2.2 Ventajas

Para comenzar con las ventajas de usar esta tecnología, tenemos el eliminar todo tipo de cables al igual que en el caso anterior. Tampoco sería necesario ningún tipo de conector. Hay que añadir lo relativamente sencillo que es crear la red inalámbrica en la que los dispositivos se sincronicen e intercambien información. El servicio es gratuito y la tecnología está masificada al igual que el WiFi, es decir, que está ampliamente aceptado.

4.2.2.1 Ley inversa del cuadrado

Como ventaja a destacar para nuestro proyecto, sería la posibilidad de calcular distancias gracias a la *ley inversa del cuadrado*. La ley inversa del cuadrado establece que, *la intensidad de cierta magnitud física es inversamente proporcional al cuadrado de la distancia a la fuente de la magnitud física*. Geométricamente hablando, la energía irradiada se esparce sobre un área que es función de la distancia al transistor. En nuestro caso sería muy útil para saber la distancia a la que se encuentra el usuario de una cámara y por consiguiente de la sala a la que tiene que ir.

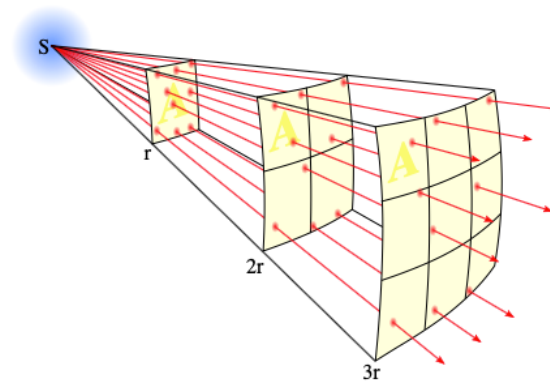


Figura 4-3. Ley Inversa del Cuadrado. Fuente: Repetidores Móviles

El cálculo aproximado sería el siguiente, cuando nos alejamos x metros, la señal disminuirá en $1/x^2$ con la siguiente relación [22]:

- A la distancia de 1 metro, tenemos cierta cantidad de potencia
- A la distancia de 2 metros, tenemos $1/4$ de la potencia que teníamos a 1 metro.
- A la distancia de 4 metros, tenemos $1/16$ de la potencia que teníamos a 1 metro.
- A la distancia de 8 metros, tenemos $1/64$ de la potencia que teníamos a 1 metro.

4.2.3 Desventajas

Uno de los principales inconvenientes que tenemos con el bluetooth, es la velocidad de las primeras versiones, recordemos que en las versiones 1.2 y 2.0 son 1Mb/s y de 3Mb/s respectivamente. Asimismo, la seguridad aquí también es un factor importante. Aunque recientemente se han presentado mejoras, tenemos el ejemplo de hace unos años, es cuando el protocolo podía llegar a resultar inseguro debido a que, con una mala configuración, era vulnerable en cuanto a la pérdida de información. Y por supuesto, no nos podemos olvidar del reducido alcance del protocolo debido, principalmente, a la baja potencia con la que trabaja.

4.3 ZigBee

ZigBee es un conjunto de protocolos de alto nivel de comunicación. Se usa para la radiodifusión digital de datos buscando ahorrar lo máximo posible en energía. Es una tecnología basada en el estándar IEEE 802.15.4. [23] Esta tecnología usa la banda ISM por lo general, adoptando la banda de 2,4GHz para comunicarse con el resto de los dispositivos, ya que es la usada por la mayoría.



Figura 4-4. Logotipo de ZigBee. Fuente: Developer Bosch IoT Suite

4.3.1 Protocolo

Es una tecnología que presume de bajo coste y reducido consumo energético. Los protocolos usados para la comunicación entre balizas o nodos hacen que se puedan entender cada miembro de la red. Unos protocolos que hacen “despertar” al nodo para comenzar con la transmisión y que, al finalizar, vuelve a dejar en reposo con el fin de ahorrar energía [23].

Aplicación / Perfil	Usuario
Soporte de Aplicación	ZigBee Alliance
Red (NWK) / Seguridad (SSP)	
MAC	IEE 802.15.4
PHY	

Tabla 4-3. Protocolos de ZigBee

El estándar seguido por los dispositivos ZigBee es el WPAN, que cuenta con una transmisión basada en la norma IEEE 802.15.4-2003. Su funcionamiento es a través de 16 canales situados en el rango de frecuencias de 2.4 GHz con un ancho de banda para cada uno de ellos de 5 MHz. El protocolo CSMA/CA que usan, tiene el fin de evitar colisiones durante la transmisión [23].

4.3.2 Topologías

Una red que cuente con dispositivos ZigBee puede tener diferentes tipologías a escoger entre estrella, árbol y malla. No obstante, la más usada es la organización en malla. Esto significa que un nodo ZigBee puede estar conectado a su vez a otros nodos más de la misma red. Con ello, se asegura la comunicación entre todos los puntos, ya que siempre existirá un camino para seguir en el caso de que alguno de ellos caiga [23].

4.3.3 Tipos de Nodos

En una red ZigBee podemos contar con diferentes dispositivos. No obstante, debemos definir los tres actores principales [23]:

- **Coordinador ZigBee.** Es el primero y mas importante de todos, ya que, si no existe coordinador, no podríamos hablar de red ZigBee. Es el nodo más completo y su misión es la de controlar toda la red y los caminos para su comunicación.
- **Router ZigBee.** Después del coordinador, nos encontramos con el router ZigBee. Se encarga de interconectar los nodos para poder ejecutar el código que un usuario haya programado. Es decir, ofrece un nivel de aplicación dentro de la torre de protocolos.
- **Dispositivo final ZigBee.** Es el último dispositivo de la red. Únicamente recibe información y solo se comunica con el nodo padre. La ventaja que tiene este dispositivo es que puede estar "dormido" y "despertarse" en ciertos momentos, con lo que alargaría la duración de su batería.

4.3.4 Ventajas

Es una tecnología que cuenta con el apoyo de diversos fabricantes de renombre como pueden ser Samsung, Bosch, Logitech o LG entre otros. También cuenta con el ahorro de batería, ya que los dispositivos se "despiertan" solo cuando es necesario [24]. Es ideal para conexiones punto a punto y punto a multipunto.

4.3.5 Desventajas

Es una tecnología nueva, que requiere de un mayor uso para saber a ciencia cierta como puede responder en ciertos momentos. La tasa de transferencia es muy baja, con lo que solo manipula textos pequeños si lo comparamos con otras tecnologías. [24] Una de las principales desventajas es que ZigBee trabaja de manera que no puede ser comparable con Bluetooth en todos sus aspectos, ya que no llegan a tener ni las mismas tasas de transferencia, ni la misma capacidad de soporte para nodos. Además,

cuenta con una menor cobertura ya que pertenece a redes inalámbricas de tipo WPAN [25].

4.4 Z-Wave

Z-Wave es un protocolo de comunicaciones inalámbricas patentado y orientado especialmente para la automatización del hogar, domótica y soluciones comerciales de control [26]. Esta tecnología hace uso de un transceptor de RF (Radio Frecuencia) de baja potencia, que se encuentra en la mayoría de equipamientos y productos de electrónica con orientación a la domótica.



Figura 4-5. Logotipo Z-Wave. Fuente: ROBOTechnics

Z-Wave se basa en una red de tipo mesh, lo que quiere decir que cada dispositivo instalado en la red se convierte en un repetidor de señal. Con lo que, cuantos más dispositivos se tengan en la red, mas fuerte será la misma. Los dispositivos de Z-Wave pueden comunicarse de punto a punto hasta 30 metros por sí solos, no obstante, tienen la capacidad de saltar señales, con lo que aseguran alcanzar fácilmente rangos efectivos de hasta 100 metros. Las redes Z-Wave tienen la capacidad de unirse con lo que se podrían lograr implementaciones más grandes. Una red Z-Wave es capaz de admitir hasta 232 dispositivos, con lo que sería factible añadir tantos dispositivos hasta asegurarse de que la red se encuentra funcionando de forma correcta [27].

Z-Wave es una red de tipo maestro/esclavo que cuenta con dos tipos de dispositivos: los controladores y los dispositivos esclavos. Los nodos pueden ser configurados para retransmitir el mensaje escuchado, con el fin de garantizar la

conectividad en el entorno con los diferentes sistemas a los que van dirigidos. El alcance medio de comunicación entre dos nodos es de aproximadamente 30.5 metros y, con la capacidad, , de “saltar mensaje” hasta cuatro veces entre los nodos a la escucha [27].

4.4.1 Ventajas

Las ventajas que presenta la tecnología Z-Wave, comienzan con la interoperabilidad con la que cuentan. Estos dispositivos se han diseñado para ser compatibles entre sí, independientemente de su tipo, fabricante, marca o versión.

Es una tecnología en crecimiento. En 2017, había alrededor de 1700 productos certificados Z-Wave en el mercado y a finales de 2019 más de 2600. Y la cifra sigue aumentando [28].

Además de su bajo consumo, otra de las ventajas de este protocolo de comunicaciones es la baja interferencia que tienen. Los dispositivos Z-Wave funcionan usando ondas de radio a la frecuencia de 800-900 MHz. Algunos de los protocolos de comunicación más comunes como pueden ser el Bluetooth y el WiFi, la frecuencia estándar es de 2,4 GHz. Que Z-Wave use una frecuencia menos habitual contribuye a reducir en una medida importante el ruido del entorno. Con lo que se consigue que el protocolo Z-Wave sea más robusto, ya que no es tan propenso a interferencias externas y posibles colisiones o accesos simultáneos al medio dentro de la red [28].

Para finalizar con las ventajas de esta tecnología, debemos destacar de nuevo la escalabilidad de estos aparatos. La mayor parte de los dispositivos Z-Wave tienen un alcance que ronda los 30 metros. No obstante, en el caso de que por necesidad se necesitase cubrir distancias mayores, se puede ampliar fácilmente el alcance de la red. Estos dispositivos pueden usarse también como repetidores, lo que conlleva que un mensaje pueda ser enviado a un dispositivo fuera del alcance del transmisor indirectamente, a través de ellos [28]. Recordemos que una red Z-Wave no solo es escalable en alcance, sino también en el número de dispositivos conectados. Pudiendo llegar la conexión de dispositivos hasta los 230, y en el caso de que hiciera falta más, se podría ampliar la red para alojar así aún más dispositivos.

4.4.2 Desventajas

Comenzamos hablando de las desventajas con una que, desde el punto de vista del autor de este trabajo, es muy importante. Y es que, aunque la tecnología esté siendo aceptada y se estén ampliando los dispositivos, es una “nueva” tecnología. Le queda camino por recorrer, y en el caso de que queramos implantarla, debemos usar equipos relativamente nuevos que cuenten con ella. Asimismo, ya que es una tecnología desarrollada con especial atención en el sector de la domótica y el internet de las cosas, los dispositivos solo admiten un parámetro en la acción que lo desencadenan. [27] Podríamos intentar añadir más equipos con el fin de realizar más de una acción, pero ya tendríamos que asumir un mayor coste y el uso de más equipos.

4.5 Conclusiones Red de Comunicación

Habiendo analizado los distintos tipos de tecnología que se podrían considerar para este proyecto. Y a falta de un simulador que refleje datos confiables donde pudiéramos diseñar un espacio para ver como se comportaría cada uno, se tendrán que atender a las especificaciones teóricas de cada tecnología para llegar a una conclusión.

Poniéndonos en situación, y dado que la persona va a acercarse al centro con su móvil, está claro que habría que elegir una tecnología que ya existiera en un teléfono móvil. Con lo que descartamos las dos últimas (ZigBee y Z-Wave), dejando las tecnologías bluetooth y WIFI como posibles candidatas. Si que podríamos valorar estas dos últimas para la conexión existente entre las cámaras que se encuentren apuntando a las pantallas y la Raspberry, ya que hay cámaras que cuentan con esta tecnología y a la Raspberry se le podría implementar una pasarela de comunicaciones, pero sería un detalle a tener en cuenta en futuras versiones.

Continuando con el análisis entre el WiFi y el Bluetooth, ambas podrían ser perfectamente válidas para el prototipo que estamos desarrollando. No obstante, llegados a este punto, debemos tener en cuenta el fin del desarrollo, el cual va dirigido hacía un público que sufre de algún tipo de discapacidad visual. Con lo que en este trabajo este punto ha sido mas determinante a la hora de decantarse por una tecnología u otra. Y la elegida es la tecnología Bluetooth, ya que ésta presenta la

posibilidad de calcular la distancia como vimos en el capítulo 4.2.2.1. Aunque por WiFi también se podría calcular, es mucho más complicado que usando Bluetooth. La idea sería que, una vez que la persona sabe hacia donde tiene que ir, pongámonos en la situación de que es una segunda planta. Cuando la persona vaya por la primera, su móvil se conectará a la cámara del primer piso y le comunicará que aún le queda un piso, pudiendo incluso decirle la distancia hasta el mismo. Una vez que siga subido hacia el segundo piso y su móvil se haya conectado a la cámara de este segundo piso, la aplicación le podrá ir comunicando cada cierto tiempo a que distancia se encuentra de su objetivo. En el caso de la existencia de otro dispositivo, se podría llegar a estudiar la posibilidad de triangular las señales para unas distancias mas próximas a la realidad.

En la figura inferior (figura 4-6), podemos ver un diagrama visual de la interconexión de los equipos que representarían el sistema. Hay que tener en cuenta que, dado que la Raspberry usada para las pruebas carece de bluetooth, el módulo de esta tecnología presentado anteriormente se encuentra conectado por USB a dicha placa.

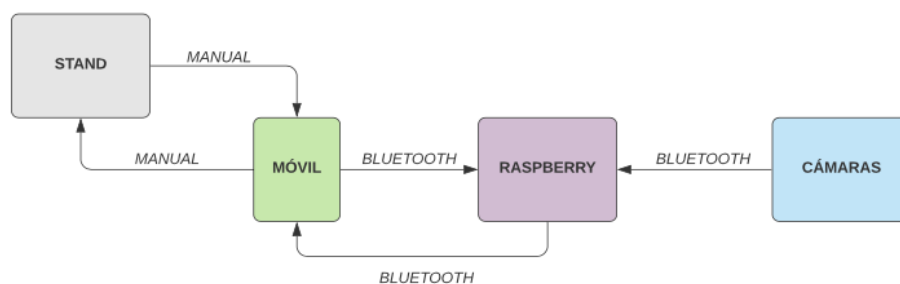


Figura 4-6. Diagrama de interconexión de los equipos

En el siguiente diagrama (figura 4-7) podemos ver de forma breve y visual los procesos que se llevan acabo entre los diferentes dispositivos.

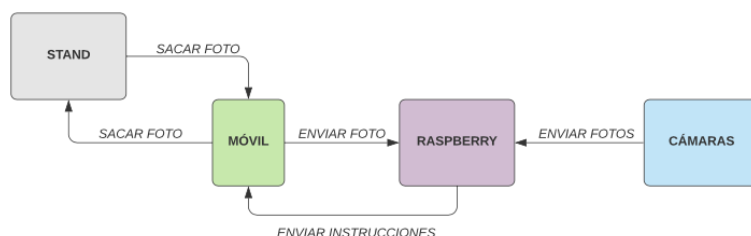


Figura 4-7. Diagrama de procesos

Capítulo 5 - Reconocimiento Óptico de Caracteres

En este capítulo se realizará un estudio del reconocimiento óptico de caracteres. Comenzando por sus inicios, como funciona, cuales han sido los modelos estudiados para este proyecto. Y finalizando con el sistema escogido, como funciona, que se podría mejorar y ejemplos obtenidos.

5.1 Introducción

El concepto de OCR [2] viene de “*Optical Character Recognition*”. Es un proceso que permite, como su propio nombre indica, el reconocimiento óptico de los caracteres contenidos en una imagen. La imagen puede ser un documento escaneado o fotografía evidentemente, y se consigue volverlos comprensibles o reconocibles al lenguaje que entiende un ordenador. Al final lo que se realiza es la digitalización de textos, los cuales se identifican a partir de una imagen, caracteres o símbolos que pertenecen a un determinado alfabeto, para luego almacenarse en forma de datos [29].

5.2 Funcionamiento

El funcionamiento general es el siguiente. En el momento que se escanea una página (ya sea impresa o escrita a mano), la misma se guarda como un archivo de mapa de bits de formato TIFF (“*Tagged Image File Format*”). Una vez realizado esto, podemos leer la imagen cuando se visualiza en la pantalla. No obstante, para el ordenador, no es “la imagen” sino que es una serie de puntos blancos y negros. Con lo que, para el ordenador, todos los textos los considera muy similares. El OCR entonces, lo que hace es inspeccionar esa imagen píxel a píxel, buscando aquellas formas que coincidan con los rasgos de los caracteres y los comparará con los que tiene en su “diccionario” o base de datos. Dependiendo de la complejidad del sistema, también podría identificar los caracteres a través del análisis de sus características, de tal forma que el reconocimiento no se limite en exclusiva a un determinado número de fuentes [30].

Aunque hoy en día el OCR tiene la capacidad de llegar a mantener la estructura de los documentos originales en el archivo de salida, e inclusive reconocer caracteres contenidos en documentos realizados a mano, partituras, etcétera. No debemos pasar por alto que hoy por hoy su nivel de efectividad sigue siendo limitado de forma considerable. No obstante, muchas aplicaciones de OCR son de pago, y el precio suele ser directamente proporcional a la calidad del software y sus resultados.

5.3 Aprendizaje Profundo

Para resolver el "problema" de OCR, a lo largo de la historia se han usado diversas técnicas de visión por ordenar, como pueden ser detectores de contornos, filtros de imagen y clasificadores de imágenes. Sin embargo, estos funcionaban bien en pequeños conjuntos de datos basados en plantillas, en el momento que variaban de forma significativa, se reducían los resultados también de forma significativa [31], produciéndose errores de generalización.

El uso de las redes neuronales profundas o Deep learning [32] ha mejorado en los últimos años, y se ha reavivado el interés de usar este con el problema de OCR. Las redes neuronales pueden jugar un papel muy importante, usándose para combinar la tarea de localizar el texto de una imagen junto con la interpretación de qué es el texto [31].

5.4 Tesseract - OCR

Tesseract – OCR es un motor de Reconocimiento Óptico de Caracteres (OCR), disponible para varios sistemas operativos. Desarrollado por Hewlett-Packard (HP) entre 1985 y 1995, en 2005 se licenció como software de código abierto y libre distribución. Sin embargo, a partir de 2006 paso a ser un desarrollo mantenido y financiado por Google y su comunidad de desarrolladores y colaboradores, aunque continúa siendo de código abierto y libre distribución [33].

5.4.1 Funcionamiento

HP desarrolló Tesseract – OCR de forma independiente la tecnología de análisis de diseño de página que se usaba en el producto, con lo que Tesseract nunca necesitó su propio análisis de diseño de página. Por ello, Tesseract asume que su entrada va a ser

una imagen binaria con regiones de texto poligonales opcionales definidas. El procesamiento de la imagen sigue un proceso tradicional paso a paso. Sin embargo, algunas de las etapas eran inusuales en el momento de desarrollo de la aplicación, y probablemente lo sigan siendo hoy en día.

El primer paso es un análisis de componentes conectados en el que se almacenan los contornos de los componentes. Aunque fue una decisión de diseño costosa en su momento (computacionalmente hablando), tenía una ventaja significativa. Y es que mediante la inspección del anidamiento de los contornos y el número de contornos de "hijos" y "nietos", es más simple detectar texto de forma inversa y reconocer de forma sencilla texto negro sobre blanco. Probablemente Tesseract fuera el primer motor de OCR con la capacidad de manejar textos en blanco sobre negro de forma tan trivial. En esta etapa, los contornos se agrupan, anidados en "blobs" (*Binary Large Object*). Los blobs son objetos binarios grandes que se usan para almacenar un elemento grande de datos en una base de datos que está en código binario. Dicho código binario es legible para el software, pero para los seres humanos solo parece una larga combinación de dos caracteres diferentes (normalmente 0 y 1) [34].

Los "blobs" se organizan en líneas de texto, y las líneas y regiones se analizan para un tono fijo o texto proporcional. Las líneas de texto a su vez se dividen en palabras de manera diferente, teniendo en cuenta el tipo de espacio entre caracteres. El texto proporcional se divide en palabras usando espacios definidos y difusos [34].

Seguidamente, el reconocimiento continúa con un proceso de dos partes. Primero, intenta reconocer cada palabra por turno. Cada palabra que se reconoce de forma satisfactoria se pasa a un clasificador adaptativo como dato de entrenamiento. Este clasificador tiene la oportunidad de reconocer con mayor precisión el texto en la parte inferior de la página. Como el clasificador adaptativo puede haber aprendido algo útil digamos que demasiado tarde para hacer una contribución cerca de la parte superior de la página, ejecuta un segundo pase sobre la misma. Con lo que, las palabras que no se reconocieron de forma correcta en la primera fase, se reconocen de nuevo en el siguiente pase. En la fase final, se resuelven espacios difusos y verifica hipótesis alternativas con la altura x de las letras con el fin de localizar el texto en minúsculas y mayúsculas [34].

5.4.2 Búsqueda de línea

El algoritmo de búsqueda de líneas es una de las pocas partes de Tesseract que previamente se han publicado [35]. Dicho algoritmo fue diseñado con el fin de poder reconocer una página sesgada sin tener que desviarla, con lo que se evita la pérdida de la calidad de imagen. Como partes principales del proceso debemos destacar el filtrado de blobs y la construcción de líneas.

Si suponemos que el análisis de diseño de página ya ha proporcionado regiones de texto de un tamaño de texto mas o menos uniforme, un filtro de altura percentil simple es capaz de “eliminar” las mayúsculas y los caracteres que se tocan verticalmente. La altura media se aproxima al tamaño del texto en la región, por ello es seguro el filtrado de los blobs que son más pequeñas que una fracción de la altura media, ya que probablemente sean signos de puntuación, signos diacríticos o ruido.

Existe una probabilidad mas alta de que los blobs filtrados se ajusten a un modelo de línea no superpuesta, es decir, paralelas pero inclinadas. El orden y procesamiento de los blobs por coordenada x hace posible el asignar blobs a una línea de texto única. Mientras se realiza el rastreo de la pendiente a lo largo de la página, el riesgo de asignar una línea de texto incorrecta con presencia de sesgo es notablemente reducido. Una vez que los blobs filtrados se han asignado a las líneas, un ajuste de la media mínima de cuadrados [36] es usada para estimar las líneas de base, y los blobs filtrados se vuelven a ajustar en las líneas apropiadas.

El último paso del proceso de creación de líneas fusiona blobs que se superponen al menos horizontalmente en la mitad, colocando las marcas diacríticas junto con la base correcta y asociando correctamente partes de algunos caracteres rotos.

5.4.2.1 Ajuste de línea de base

Una vez que se han encontrado las líneas de texto, las líneas de base se ajustan con mayor precisión usando una spline (curva diferenciable definida en porciones mediante polinomios) cuadrática. Esto fue una novedad para un sistema OCR que permitió a Tesseract el manejo de página con líneas base curvas [37], algo muy común en el escaneo que no solo ocurre en libros ya encuadernados.

Figura 5-1. Ejemplo de línea de base ajustada a curva. Fuente: Optical Character Recognition

Las líneas de base se ajustan dividiendo los blobs en grupos con un desplazamiento razonablemente continuo de la línea de base recta original. La spline cuadrática se ajusta a la partición que se encuentre más poblada (se supone que es la línea de base), haciendo uso de un ajuste de mínimos cuadrados. El spline cuadrático cuenta con la ventaja de que este cálculo es razonablemente estable, aunque cuenta con la desventaja de que pueden surgir discontinuidades cuando se requieren múltiples segmentos de spline. Probablemente una spline cúbica más tradicional podría funcionar mejor [38].

En la figura anterior (figura 5-1), podemos ver un ejemplo de una línea de base ajustada, una línea descendente, una línea media y una línea ascendente. Podemos apreciar que todas estas líneas son paralelas, (ya que la separación en el eje "y" es una constante en toda la longitud), y ligeramente curvadas. La línea ascendente es azul y la línea negra de encima es en realidad recta.

5.4.2.2 Detección y corte de tono fijo

Tesseract realiza pruebas con las líneas de texto para determinar si son de tono fijo. Cuando encuentra dicho texto de tono fijo, Tesseract divide las palabras en caracteres usando el tono y deshabilita el "interruptor" y el asociador en estas palabras para el paso de reconocimiento de palabras [34].

5.4.2.3 Búsqueda proporcional de palabras

El espaciado de texto no fijo o proporcional es un trabajo que dista mucho de ser trivial. En la figura 4-2 podemos observar algunos de los problemas más típicos. Observamos que el espacio entre las unidades y decenas de "11.9%" es de un tamaño similar al del espacio general. No obstante, es un tamaño mayor que existe entre las palabras "erated" (errado en inglés) y "junk" (baura en inglés). No existe tampoco ningún tipo de espacio horizontal entre los cuadros delimitadores de "of" (de en inglés) y "financial" (financiero en inglés) [34].

**of 9.5% annually while the Fed-
erated junk fund returned 11.9%
*fear of financial collapse,***

Figura 5-2. Dificultades de espaciado de texto. Fuente: Optical Character Recognition

Tesseract es capaz de resolver la mayor parte de estos problemas, midiendo los espacios en un rango vertical limitado entre la línea de base y la línea media. Los espacios que se encuentran más cerca del umbral en esta etapa se vuelven borrosos, de modo que se puede tomar una decisión final después del reconocimiento de palabras [34].

5.4.3 Reconocimiento de palabras

Una parte importante del proceso de reconocimiento de cualquier motor de reconocimiento de caracteres es el de identificar cómo se debe de segmentar una palabra en caracteres. En primer lugar, se clasifica la salida de segmentación inicial de la búsqueda de línea. El resto de los pasos del reconocimiento de palabras se aplica únicamente al texto de paso no fijo [34].

5.4.3.1 Troceando caracteres unidos

En el caso de que el resultado de una palabra no es del todo satisfactorio, Tesseract intenta mejorar el resultado cortando el blob con el peor resultado del clasificador de caracteres. Los candidatos a puntos de corte se encuentran a partir de vértices cóncavos de una aproximación poligonal [39] del contorno, y pueden tener otro vértice cóncavo opuesto o un segmento de línea. Puede llevar hasta tres pares de puntos de corte el separar con éxito los caracteres unidos del conjunto ASCII.

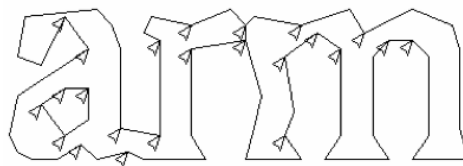


Figura 5-3. Candidatos de puntos de corte y corte. Fuente: The Extraction and Recognition of Text from Multimedia Document Images

En la figura superior (figura 4-3), podemos ver un conjunto de puntos de corte candidatos con flechas y el corte seleccionado como una línea a través del contorno donde la letra “r” se junta con la “m”.

Los cortes se ejecutan en orden de prioridad. Cualquier corte que no mejore la confianza del resultado se deshace, aunque no se descarta por completo con el fin de que el asociador tenga la posibilidad de reutilizarlo más tarde en el caso de que fuese necesario.

5.4.3.2 Asociación de caracteres rotos

En el momento en el que los cortes potenciales se han agotado, y si la palabra aún no es lo suficientemente buena, ésta se le da al asociador. El asociador realiza una búsqueda “A*” (lo mejor primero) del gráfico de segmentación de posibles combinaciones de los blobs máximos cortados en los caracteres candidatos. Lo realiza sin construir realmente el gráfico de segmentación, sino que mantiene una tabla hash de aquellos estados visitados. La búsqueda “A*” continúa sacando nuevos estados candidatos de una cola prioritaria y evaluándolos, clasificando aquellas combinaciones no clasificadas de fragmentos.

Se puede llegar al razonamiento que este enfoque de realizar el corte y a continuación asociar es, suponiendo el mejor de los casos, ineficiente y, en el peor de los casos, que provoque una pérdida importante de cortes, y puede llegar a darse el caso. La ventaja que tienes, es que el esquema de cortar y seguidamente asociar simplificar las estructuras de datos que serían necesarias para conseguir mantener el gráfico de segmentación completo.

En 1989 que se implementó por primera vez este sistema de búsqueda de segmentación “A*”, la precisión que demostraba Tesseract en caracteres rotos se encontraba muy por delante de los motores comerciales de la época. En la figura siguiente (figura 4-4), podemos ver un ejemplo típico. Una parte esencial de ese éxito fue el clasificador de caracteres, que podía reconocer con facilidad los caracteres rotos.

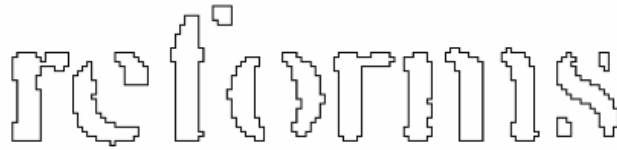


Figura 5-4. Palabra fácilmente reconocible. Fuente: *The Extraction and Recognition of Text from Multimedia Document Images*

5.4.4 Clasificador de caracteres estático

5.4.4.1 Características

Una de las primeras versiones de Tesseract uso características topológicas desarrolladas a partir del trabajo de Shillman et. Al. [40] [41] Aunque son bastante independientes de la fuente y el tamaño, dichas características no son sólidas para los problemas encontrados en las imágenes de la vida real [42]. Una idea intermedia conllevó el uso de segmentos de la aproximación poligonal como características, aunque este enfoque tampoco es robusto para los caracteres dañados. Si de la siguiente figura (figura 4-5), nos fijamos en la "b", el lado derecho del eje se encuentra en dos piezas principales, pero en la "a" observamos una única pieza.

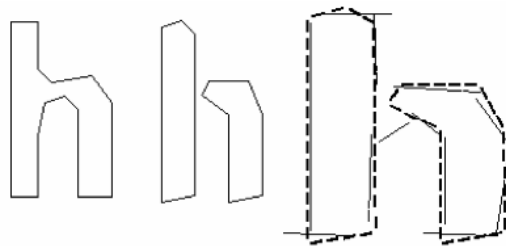


Figura 5-5. (a) "h" prístina, (b) "h" rota, (c) característica empareja con prototipos. Fuente: *Omnidocument Technologies*

La solución innovadora es la idea de que las características de lo desconocido no tienen por qué ser las mismas que las características de los datos de entrenamiento. Durante el entrenamiento, los segmentos de una aproximación poligonal se usan para las características [39], sin embargo, en reconocimiento, las características de una longitud pequeña y fija (atendiendo a unidades normalizadas) se extraen del contorno y se combinan de uno a uno con las características del prototipo agrupado de los datos de entrenamiento. Volviendo a la anterior figura (figura 4-5), nos fijamos en la c, las líneas cortas y mas anchas son las características que se han extraído de lo desconocido. Y las

líneas mas largas y estrechas son los segmentos agrupados de la aproximación poligonal que se usan como prototipos. Un prototipo que une las dos piezas es completamente inigualable. Tres características en un lado y dos en el otro son incomparables, aunque, aparte de eso, cada prototipo y cada característica se encuentra bien combinados. Con este ejemplo somos capaces de ver que este proceso de características pequeñas que coinciden con prototipos grandes es fácilmente capaz de hacer frente al reconocimiento de imágenes dañadas. Aunque el principal obstáculo es que el costo computacional de calcular la distancia entre una incógnita y un prototipo es considerablemente elevado.

Recapitulando, las características que se extraen de lo desconocido son, por lo tanto, tridimensionales (posición x, posición y e ángulo), con entre cincuenta y cien características en un carácter. Las características del prototipo son cuatro dimensiones (x-y, posición, ángulo y longitud), con características típicas de entre diez y 20 en una configuración prototipo.

5.4.4.2 Clasificación

Si hablamos de la clasificación, es un proceso en dos pasos. En el primer paso, un podador de clases crea una lista corta de clases de personajes que podría coincidir con el "desconocido". Cada característica obtiene, de una tabla de búsqueda tridimensional cuantificada de forma aproximada, un vector de bits de clases que podría coincidir, y los vectores de bits se suman sobre todas las características. Las clases con los recuentos más altos (una vez corregido el número esperado de características), se convierten en la lista corta para el siguiente paso.

Cada característica de lo desconocido busca un vector de bits de prototipos de la clase dada que podría coincidir, y seguidamente se calcula la similitud real entre ellos. Cada clase de carácter prototipo se encuentra representada por una expresión lógica de suma de productos con cada término llamado configuración, por lo que el proceso de cálculo de distancia mantiene un registro de la evidencia de similitud total de cada característica en cada configuración, así como de cada prototipo. La mejor distancia combinada calculada a partir de la característica sumada y las evidencias del prototipo es la mejor sobre todas las configuraciones almacenadas de la clase.

5.4.4.3 Datos de entrenamiento

Desde que el clasificador es capaz de reconocer fácilmente los caracteres dañados, el clasificar no ha sido entrenado en los caracteres dañados. De hecho, el clasificador fue entrenado con veinte muestras de noventa y cuatro caracteres de ocho fuentes en un solo tamaño, pero con cuatro atributos (normal, negrita, cursiva y negrita cursiva), lo que hace un total de sesenta mil ciento sesenta muestras de entrenamiento. Esto es un contraste muy significativo comparado con otros clasificadores, como el clasificador Caleta que cuenta con más de un millón de muestras y el clasificador de cien fuentes de Baid, con un millón ciento setenta y cinco mil muestras de entrenamiento.

5.4.5 Análisis Lingüístico

Tesseract contiene muy poco análisis lingüístico. En el momento en el que el módulo de reconocimiento de palabras se encuentre considerando una nueva segmentación, el módulo lingüístico (denominado erróneamente "permuter"), elige la mejor cadena de palabras disponible de acuerdo con las siguientes categorías:

- Palabra más frecuente
- Palabra del diccionario más frecuente
- Palabra que más veces ha aparecido
- Palabra mayúscula más frecuente
- Palabra minúscula más frecuente (con minúscula inicial opcional)
- Palabra más seleccionada por el clasificador

La decisión final para una segmentación determinada es simplemente la palabra con la clasificación de distancia total más baja, donde cada una de las categorías anteriores se multiplica por una constante diferente.

Las palabras de diferentes segmentaciones pueden tener diferentes números de caracteres. Es complicado comparar las palabras directamente, incluso cuando un clasificado afirma estar creando probabilidad, algo que Tesseract no hace. Dicho problema en Tesseract se resuelve generando dos números para cada clasificación de

caracteres. El primero, llamado de seguridad o confianza, es la menor distancia normalizada del prototipo. Lo que permite que sea una "confianza" en el sentido de que números mayores son mejor, pero todavía es una distancia, ya que cuanto mas lejos se encuentre de cero, mayor será la distancia. La segunda salida, conocida como "clasificación", multiplica la distancia normalizada desde el prototipo por la longitud total del contorno en el carácter desconocido. Las calificaciones obtenidas de los caracteres dentro de una palabra se pueden sumar de forma significativa, ya que la longitud total del contorno de todos los caracteres dentro de una palabra es siempre la misma.

5.4.6 Clasificador adaptativo

Se ha investigado [43] y demostrado [44] que los motores OCR pueden beneficiarse del uso de un clasificador adaptativo. Dado que el clasificador estático tiene que ser bueno para poder usarlo con cualquier tipo de fuente, su capacidad para discriminar entre diferentes caracteres o entre caracteres y no caracteres se debilita. Por lo tanto, se usa comúnmente un clasificador adaptativo más sensible a las fuentes [45] que es entrenado por la salida del clasificador estático para obtener una mayor discriminación dentro de cada documento, ya que el número de fuentes es limitado.

Tesseract no hace uso de un clasificador de plantilla, sin embargo, usa las mismas características y clasificador que el clasificador estático. La única diferencia que existe entre el clasificador estático y el clasificador adaptativo, además de los datos de entrenamiento, la encontramos en que el clasificador adaptativo usa la normalización isotrópica de línea base/altura x . Mientras tanto, el clasificador estático por su parte normaliza los caracteres por el centroide (primeros momento), para la posición y el segundo momento para la normalización del tamaño anisotrópico.

La normalización de la línea de base/altura x facilita la distinción de caracteres en mayúsculas y minúsculas, además de mejorar la inmunidad frente a las manchas de ruido. El principal beneficio de la normalización del momento del carácter se encuentra en la eliminación de la relación de aspecto de la fuente y cierto grado de ancho de trazo de la fuente. También simplifica el reconocimiento de subíndices y superíndices,

aunque requiere una función de clasificador adicional para distinguir algunos caracteres en mayúsculas y minúsculas.



*Figura 5-6. Letras en forma normalizada de línea de base/altura x y forma normalizada de momento.
Fuente: Automatic Prototype Extraction for Adaptive OCR*

En la anterior figura (figura 4-6), podemos observar un ejemplo de tres letras en forma normalizada de línea de base/altura x y forma normalizada de momento.

5.5 Conclusiones OCR

Finalmente se ha optado por el uso de la aplicación Tesseract entre otras cosas por sus resultados y porque es de código abierto. En el máster de internet de las cosas nos han enseñado que usar programas de código abierto tiene muchas ventajas sobre los que no lo son, sobre todo porque te permite usar distintos elementos de diferentes marcas o casas sin tener que “casarte” con una u otra. Esto es muy útil ya que, a la hora de diseñar un sistema o despliegue, se podrían usar elementos que consideremos mejores por las razones oportunas de diferentes marcas trabajando en conjunto para nuestros fines.

Capítulo 6 - Aplicación

A la hora de comenzar con el desarrollo se dudaba entre los sistemas Apache Cordova [46] y Android Studio [14]. Finalmente me decanté por este último, ya que se podría hacer las pruebas de forma mas sencilla y probar la aplicación en un Smartphone Android con el que se contaba. Aunque el desarrollo de la aplicación supuso un reto para el autor, en la web se pueden encontrar múltiples cursos e información para llegar a realizar una aproximación de la aplicación. El código Kotlin con el que se ha desarrollado la aplicación es el lenguaje que Android quiere que sus desarrolladores acaben usando, frente a Java.

6.1 Interfaz (Front End)

Primero, y como es lógico, el usuario tendría que realizar una foto del tique. Por ello, lo primero que debemos hacer en la aplicación es sacar una foto de forma sencilla.

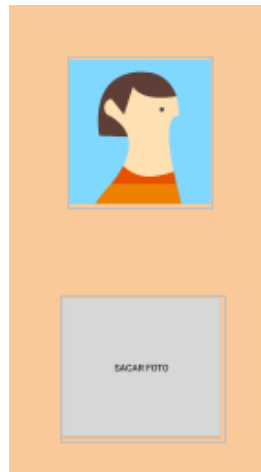


Figura 6-1. imagen de desarrollo de la aplicación

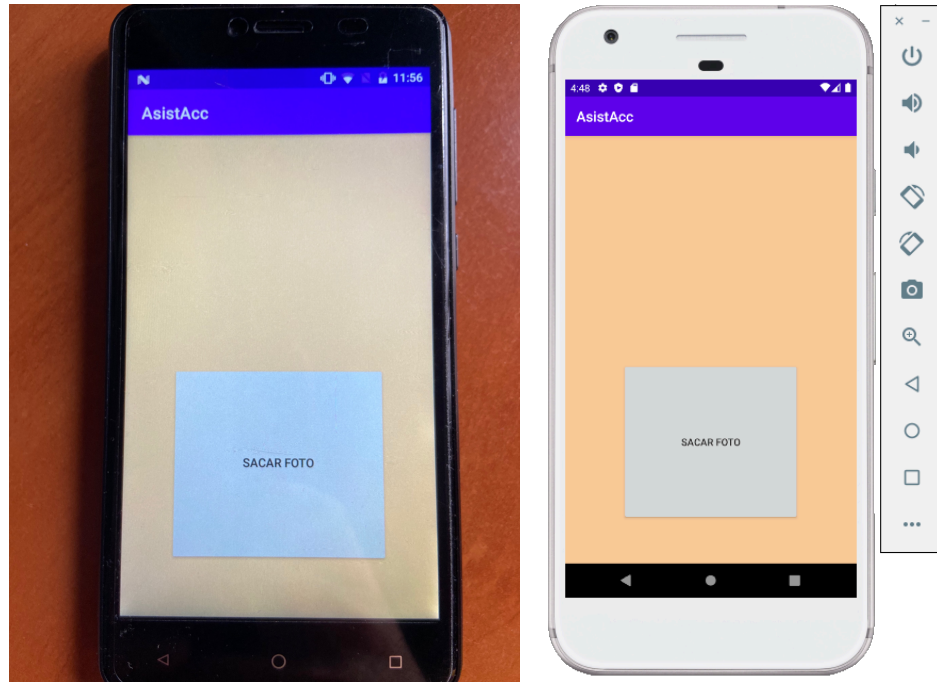


Figura 6-2. Foto de la aplicación (Just5 M503 y Google Pixel Simulado)

En el momento que abrimos la aplicación nos encontramos con lo que podemos observar en la anterior figura (figura 5-2). Dado que es una aplicación para personas que tengan algún tipo de carencia visual, ha de ser muy simple e intuitiva, con lo que únicamente tendrá un botón bastante amplio en la zona inferior, que al ser presionado abrirá la cámara automáticamente.

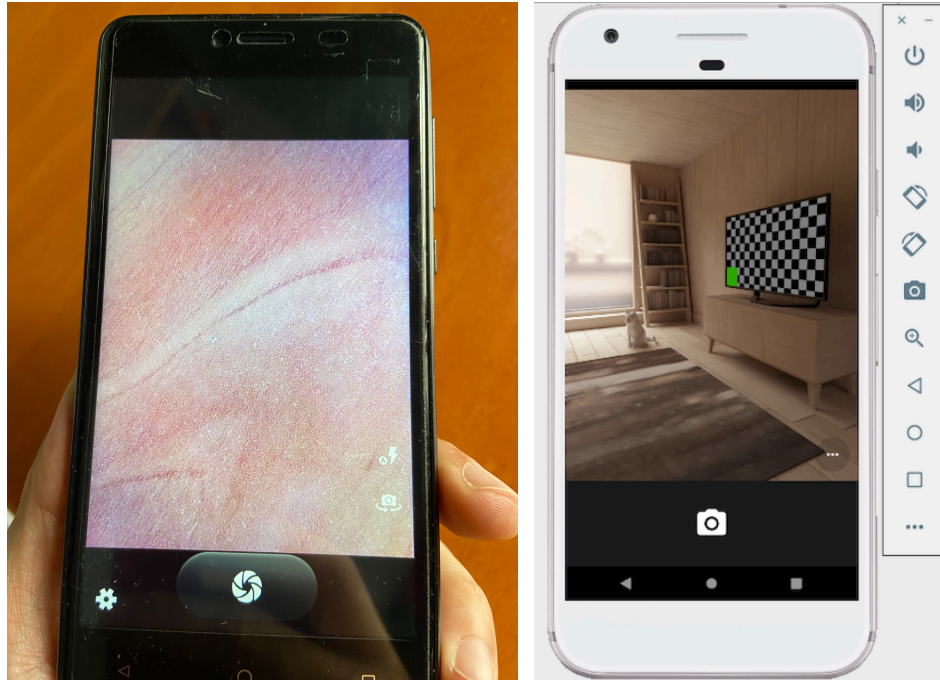


Figura 6-3. Al presionar el botón, nos abre la aplicación de la cámara (Just5 M503 y Google Pixel Simulado)

Al sacar la foto, dependiendo de la aplicación de la cámara del móvil, nos aparecerá para aceptar o repetir la foto.

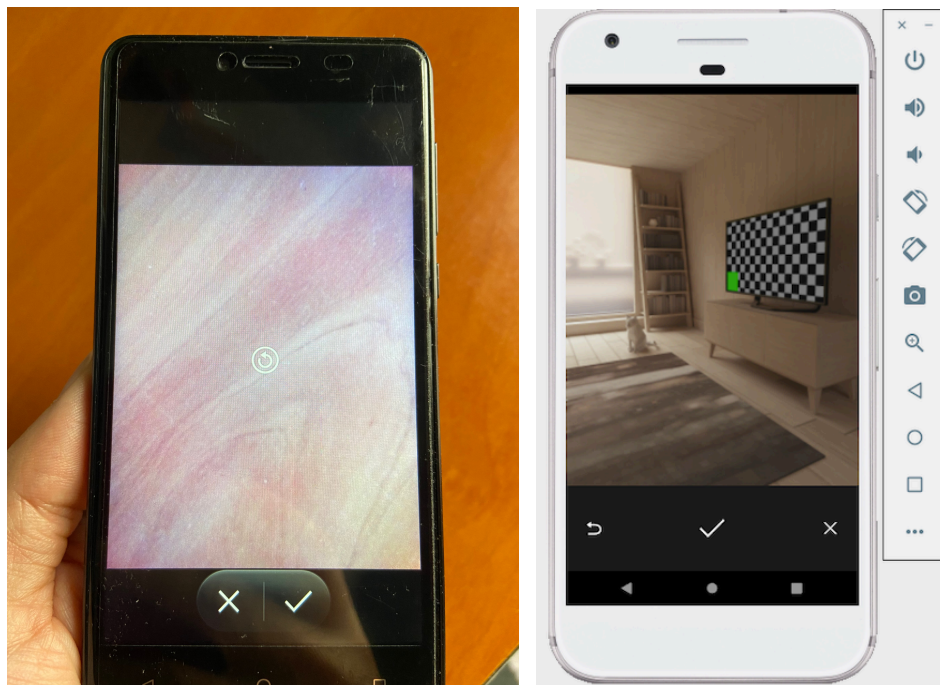


Figura 6-4. Una vez sacada la foto, debemos aceptarla (Just5 M503 y Google Pixel Simulado)

Y una vez aceptada, nos aparecerá una imagen de vista previa encima del botón de la pantalla principal de la aplicación.

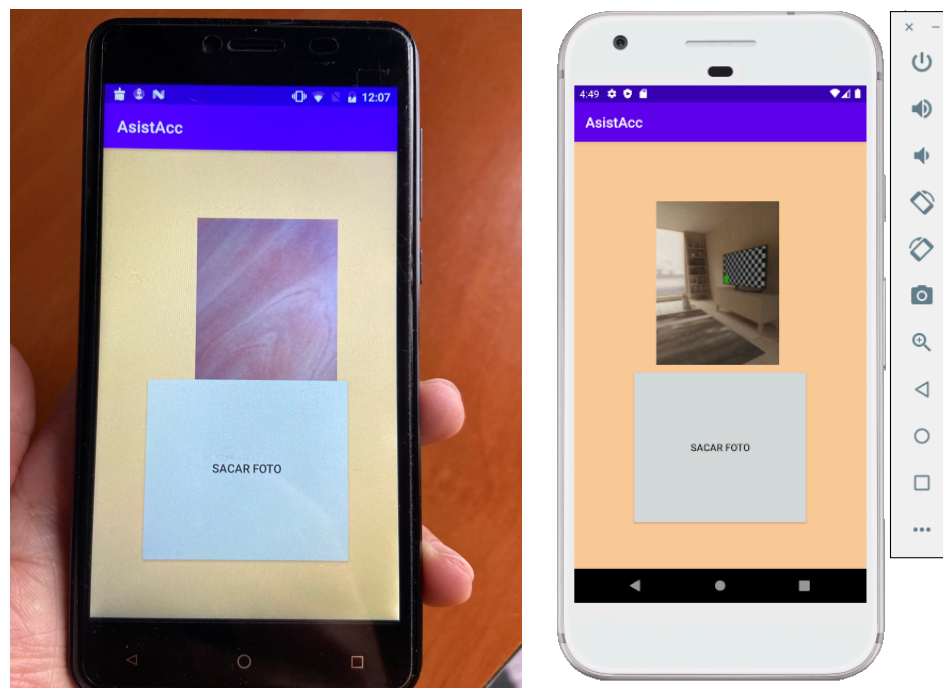


Figura 6-5. Vista previa de la foto (Just5 M503 y Google Pixel Simulado)

En el momento que se tenga respuesta de la Raspberry con las instrucciones, la aplicación cambiará a una pantalla de notificaciones, desde la que se irá guiando al usuario hasta su destino. En la pantalla solo aparecerá texto, sin embargo, el usuario puede seguir haciendo uso de *TalkBack* con el fin de poder seguir las indicaciones.

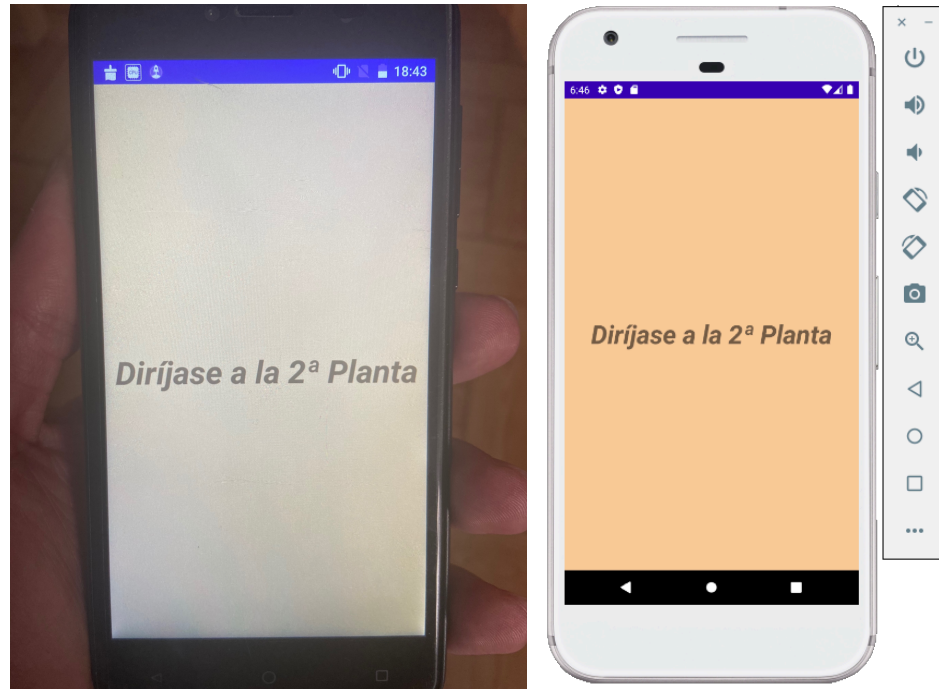


Figura 6-6. Pantalla de notificaciones (Just5 M503 y Google Pixel Simulado)

6.2 Back End

Lo que envuelve al entorno gráfico que acabamos de ver, en una programación realizada en Kotlin.

En Android STUDIO cada objeto o componente lleva una programación por detrás, o, mejor dicho, una programación es lo que hace que exista un objeto o componente.

Hay dos ubicaciones o archivos importantes a la hora de programar una aplicación con Android STUDIO, el `activity_main.xml` y el `MainActivity.kt`.

6.2.1 Activity_main

En este apartado es donde se ubica la programación gráfica de la aplicación, es decir, todo lo que tiene que ver con colores, ubicaciones, textos, etc.

En esta parte se ha programado lo que sería la pantalla principal con la que comenzar a trabajar, el "fondo" para denominarlo de alguna forma.

Seguidamente se ha continuado programando el botón (componente "Button"). A la hora de añadir estos componentes, se ha de asignar un id, mediante el cual son identificados de manera inequívoca en tareas futura. Asimismo, también se ha referenciado en pantalla, tanto vertical como horizontalmente.

A continuación, se añadió el campo para ver la imagen una vez realizada, usando el componente "ImageView". También se tuvo que añadir un id y referenciarlo en pantalla, horizontal y verticalmente al igual que el anterior objeto.

6.2.2 MainActivity

En el archivo MainActivity.kt es donde se programa las funcionalidades de los objetos definidos anteriormente. Para esta aplicación, se ha hecho uso de una clase donde se han programado las acciones. Por un lado, el botón "SACAR FOTO" que, tras darle permisos, abre la cámara. Una vez realizada la foto, se almacena con el fin de terminar mostrándola en el otro objeto.

6.2.3 Notifications

Este apartado cuenta con toda la programación a nivel visual y de código de la pantalla de notificaciones y su conexión con las Raspberry.

Cuenta con un *TextView* desde el que se puede asociar a un *string* con el fin de mostrar las indicaciones que lleguen desde la Raspberry.

6.3 Accesibilidad de la aplicación

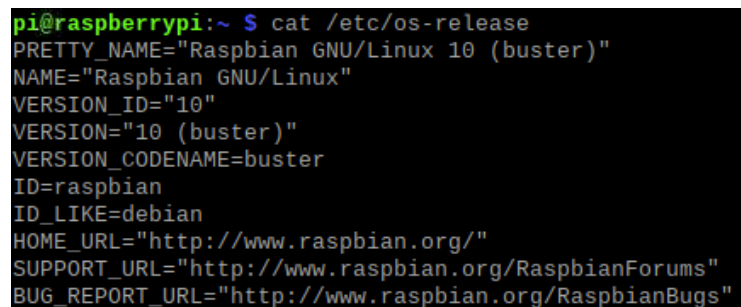
Según la definición de Apple: *“Una aplicación es accesible cuando todos los elementos de la interfaz de usuario con los que los usuarios pueden interactuar son accesibles. Un elemento de la interfaz de usuario es accesible cuando indica correctamente que es un elemento de accesibilidad.”* [47]

Ateniéndonos a esta definición, ¿podemos afirmar que la aplicación de prueba desarrollada cumple los estándares de accesibilidad? Como hemos visto en las imágenes superiores, la aplicación cuenta con un simple botón de proporciones importantes que comunica con la aplicación de la cámara del dispositivo. Si bien no se le ha insertado audio, ya que se puede hacer uso del asistente TalkBack [3] en los dispositivos Android como de VoiceOver [5] en los dispositivos iOS. Con lo que, en este apartado, se cumple el estándar de accesibilidad en la aplicación.

Capítulo 7 - Raspberry Pi

La Raspberry pi en este desarrollo es el centro de control de las operaciones. Aún sabiendo que el tema del reconocimiento óptico de caracteres se podría haber realizado íntegramente en la aplicación, ocuparía mucho más espacio y tampoco sería viable debido a que cada usuario tiene un tipo diferente de dispositivo con diferente capacidad de procesamiento.

Como sistema operativo de la Raspberry he usado un Raspbian versión 10.



```
pi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 10 (buster)"
NAME="Raspbian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
```

Figura 7-1. Versión sistema operativo Raspberry

Debemos diferenciar dos procesos en la Raspberry pi:

- a) los que corresponderían al reconocimiento óptico de caracteres
- b) la recepción de las imágenes por bluetooth.

7.1 Reconocimiento óptico de caracteres

Para el reconocimiento óptico de caracteres, como ya vimos en el capítulo 4, he hecho uso de la aplicación de Tesseract.

```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get update  
Obj:1 http://raspbian.raspberrypi.org/raspbian buster InRelease  
Obj:2 http://archive.raspberrypi.org/debian buster InRelease  
Leyendo lista de paquetes... Hecho  
pi@raspberrypi:~$ sudo apt-get install tesseract-ocr  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.  
rpi-eeeprom-images  
Utilice «sudo apt autoremove» para eliminarlo.  
Se instalarán los siguientes paquetes adicionales:  
  libliblept5 libtesseract4 tesseract-ocr-eng tesseract-ocr-osd  
Se instalarán los siguientes paquetes NUEVOS:  
  libliblept5 libtesseract4 tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd  
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
Se necesita descargar 6.651 kB de archivos.  
Se utilizarán 20,9 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] s  
Des:1 http://ftp.cica.es/mirrors/Linux/raspbian/raspbian buster/main armhf libl  
pt5 armhf 1.76.0-1 [804 kB]  
Des:2 http://ftp.cica.es/mirrors/Linux/raspbian/raspbian buster/main armhf libt  
esseract4 armhf 4.0.0-2 [1.032 kB]
```

Figura 7-2. Instalación Tesseract en la Raspberry Pi

En mi caso he instalado la versión 4.0, con los idiomas de español, inglés y OSD (On Screen Display), que se pueden ver en la siguiente figura (figura 7-3).

```
pi@raspberrypi:~$ tesseract --list-langs  
List of available languages (3):  
eng  
osd  
spa
```

Figura 7-3. Lista de idiomas instalados de Tesseract

Hay dos formas de ejecutar Tesseract, por medio de terminal o gráficamente. En mi caso lo he usado por terminal, ya que gráficamente da mas problemas. Para ejecutarlo, basta con ejecutar el siguiente comando:

Tesseract *nombre_imagen* *nombre_txt_salida* -l *idioma*

Donde:

- **Nombre_imagen** es el nombre que tiene la imagen de entrada de la que queremos realizar el reconocimiento óptico de caracteres.
- **Nombre_txt_salida** es el nombre del archivo de salida. La salida que da Tesseract una vez ejecutado es un txt con el resultado de la transcripción.
- **Idioma** es el idioma en el que introducimos la imagen para que el programa use un diccionario u otro, en mi caso he usado tanto español como inglés. No obstante, dado que lo que se necesita transcribir no tiene un idioma en concreto porque son letras y números aleatorios sin ningún tipo de significado, consigue detectar lo mismo independientemente del idioma seleccionado.

```

pi@raspberrypi:~ $ tesseract pantalla6.jpg pantalla6.jpg -l spa
Tesseract Open Source OCR Engine v4.0.0 with Leptonica
Warning: Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 709
Detected 9 diacritics
pi@raspberrypi:~ $ tesseract pantalla7.jpg pantalla7.jpg -l spa
Tesseract Open Source OCR Engine v4.0.0 with Leptonica
Warning: Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 299
Detected 9 diacritics
pi@raspberrypi:~ $ tesseract ticket1.png ticket1.png -l spa
bash: error sintáctico cerca del elemento inesperado `('
pi@raspberrypi:~ $ tesseract ticket1.png ticket1 -l spa
Tesseract Open Source OCR Engine v4.0.0 with Leptonica
Warning: Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 294

```

Figura 7-4. Pruebas ejecución de Tesseract

El funcionamiento de Tesseract es sencillo, en este caso se ha ejecutado en la carpeta donde se encontraban las imágenes para hacer las pruebas de forma mas sencilla disminuyendo las probabilidades de equívoco.

7.2 Recepción de las imágenes por bluetooth

Para la recepción de las imágenes por Bluetooth, como ya he comentado anteriormente, dado que mi modelo de Raspberry no viene con dicha tecnología, he usado un módulo bluetooth por USB. Y para hacer uso de dicho módulo, he instalado la aplicación “blueman”.

```

pi@raspberrypi:~ $ sudo apt-get update
Des:1 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:2 http://archive.raspberrypi.org/debian buster InRelease [32,6 kB]
Des:3 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0 MB]
Des:4 http://archive.raspberrypi.org/debian buster/main armhf Packages [330 kB]
Descargados 13,4 MB en 28s (472 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~ $ sudo apt-get install pi-bluetooth blueman
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
pi-bluetooth ya está en su versión más reciente (0.1.13).
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
rpi-eeeprom-images
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
bluez-obexd gir1.2-appindicator3-0.1 gir1.2-notify-0.7 libappindicator3-1

```

Figura 7-5. Instalación Blueman

Una vez realizado esto, he tenido que emparejar el móvil a la Raspberry para poder enviar las imágenes. Es algo que habría que mejorar en el caso de versiones futuras, ya que tendría que ser de forma automática, sin que el usuario tuviera que realizar nada.

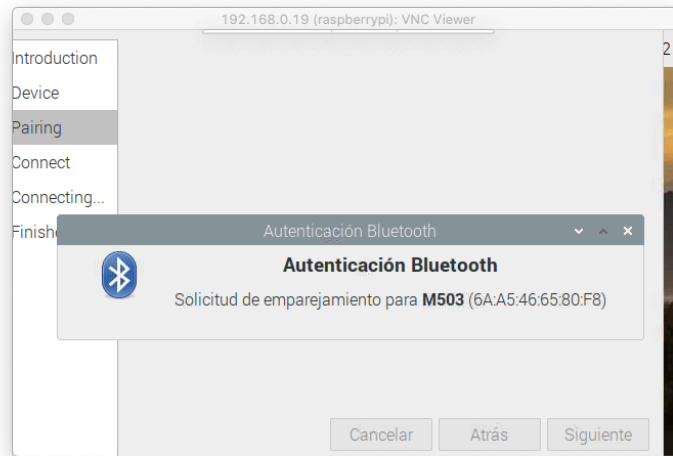


Figura 7-6. Emparejamiento por bluetooth (1/3)

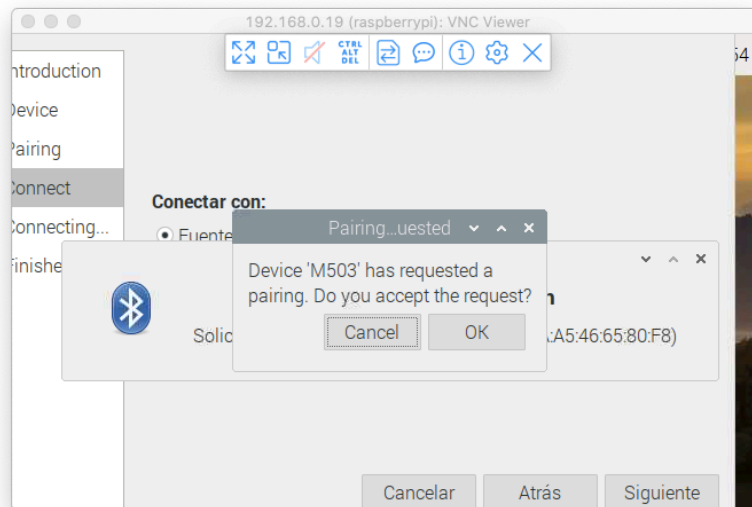


Figura 7-7. Emparejamiento por bluetooth (2/3)

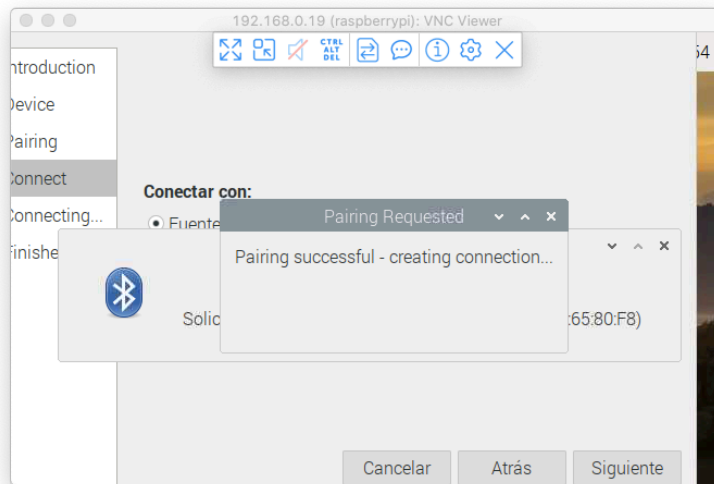


Figura 7-8. Emparejamiento por bluetooth (3/3)

7.3 Envío de notificaciones a la aplicación

Una vez analizado el texto reconocido de la imagen, que dependerá del tipo de tique con el que cuente, se comenzarán a enviar notificaciones al usuario a través de la aplicación con las indicaciones que ha de seguir.

Para este envío, se habrían de enviar por la misma tecnología que se ha recibido la foto, es decir, sockets vía bluetooth que la aplicación sea capaz de leer y mostrárselos al usuario. El código [48] en la Raspberry tendría un aspecto similar al siguiente:

```
import bluetooth

bd_addr = "88:A7:3C:21:DC:1E"

port = 1

sock=bluetooth.BluetoothSocket( bluetooth.RFCOMM )

sock.connect((bd_addr, port))

sock.send("Diríjase a la 2ª Planta ")

sock.close()
```


En la aplicación sería necesario programar a la aplicación para que fuera leyendo los strings que le llegan a través del puerto que enviáramos e ir mostrándolos en la pantalla de notificaciones.

Capítulo 8 - Pruebas Realizadas

8.1 Introducción

En este capítulo veremos las pruebas realizadas usando el reconocimiento óptico de caracteres. El escenario ha sido el siguiente, se han transferido las imágenes a la Raspberry, y una vez en dicho sistema, se ha procedido a usar la herramienta Tesseract por las razones que hemos visto en el capítulo cinco.

El banco de imágenes con las que se han realizado las pruebas se ha obtenido de internet debido a que, por la situación que hemos pasado, no ha sido posible obtenerlas de forma presencial.

8.2 Parámetros Tesseract

Tesseract tiene 678 valores configurables. Estos parámetros se encuentran en un archivo de texto sin formato y son pares clave/valor delimitadas por espacios, cada una de ellas en una línea separada por punto y coma. Para las presentes pruebas se han dejado por defecto.

8.3 Pruebas

A continuación, se muestran los resultados de las pruebas realizadas. Las pruebas se han realizado directamente en la Raspberry usando la aplicación de Tesseract. El archivo de entrada corresponde al archivo de imagen que se "introduce" en la aplicación y el archivo de salida al txt con el texto que se ha sacado de dicha imagen.

- **Imagen 1**

Archivo de entrada:



Figura 8-1. Archivo de entrada 1. Fuente: EnfermeriaDeCiudadReal

Archivo de salida:

Sin resultados.

El formato de la imagen de entrada ha sido jpeg. Sin embargo, el resultado que arroja la imagen es una página en blanco. No detecta nada.

- **Imagen 2**

Archivo de entrada:

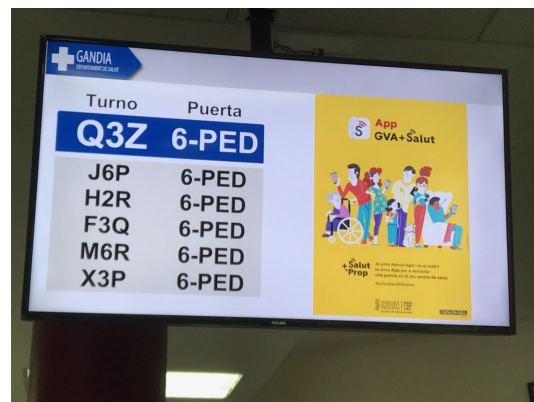


Figura 8-2. Archivo de entrada 2. Fuente: Gandía Departament de Salut

Archivo de salida:

Sin resultados.

El formato de la imagen de entrada ha sido jpg. Sin embargo, el resultado que arroja la imagen es una página en blanco. No detecta nada.

- **Imagen 3**

Archivo de entrada:



Figura 8-3. Archivo de entrada 3. Fuente: Sanidad Castilla La Mancha

Archivo de salida:

Sin resultados.

El formato de la imagen de entrada ha sido jpg. Sin embargo, el resultado que arroja la imagen es una página en blanco. No detecta nada.

- **Imagen 4**

Archivo de entrada:



Figura 8-4. Archivo de entrada 4. Fuente: Change.org

Archivo de salida:

AS

Turno Consulta

```
P8c
> L2Kk 30
: R9D 30

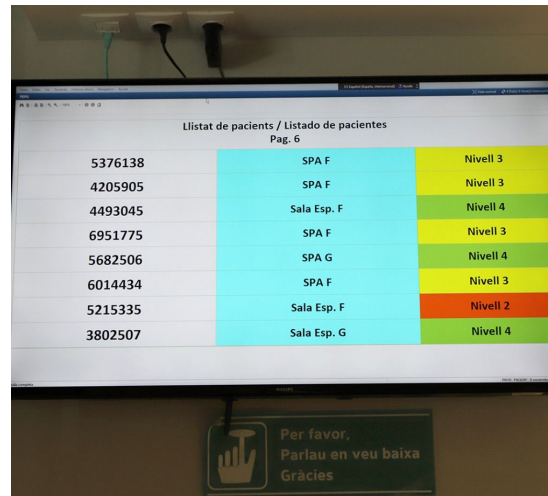
2 W7Z 29
B5Z
```

Figura 8-5. Archivo de salida 4

El formato de la imagen de entrada ha sido png. Podemos observar que es capaz de transcribir algunas partes de la imagen.

- **Imagen 5**

Archivo de entrada:



Llistat de pacients / Listado de pacientes		
Pag. 6		
5376138	SPA F	Nivell 3
4205905	SPA F	Nivell 3
4493045	Sala Esp. F	Nivell 4
6951775	SPA F	Nivell 3
5682506	SPA G	Nivell 4
6014434	SPA F	Nivell 3
5215335	Sala Esp. F	Nivell 2
3802507	Sala Esp. G	Nivell 4

Figura 8-6. Archivo de entrada 5. Fuente: 20minutos

Archivo de salida:

```

A

5376138
4205905
4493045
6951775
5682506
6014434
5215335
3802507

Llistat de pacients / Listado de pacientes
Pag. 6
Nivell 3
SPA F
"Sala Esp. F
SPAF Nivell 3

Nivell 3

Nivell 2

```

Figura 8-7. Archivo de salida 5

El formato de la imagen de entrada ha sido jpg. Vemos que en este caso es capaz de identificar todos los datos de la primera columna, e incluso la cabecera de la tabla. Sin embargo, de las otras dos columnas apenas es capaz de identificar un par de datos.

- **Imagen 6**

Archivo de entrada:

NÚMERO	MESA
A61	1
A60	1
A59	1

Figura 8-8. Archivo de entrada 6. Fuente: GESTORN

Archivo de salida:

```
O g6A5

TroGo. Fecha al o
AG 1

A60 1
```

Figura 8-9. Archivo de salida 6

El formato de la imagen de entrada ha sido jpg. En este caso, siendo una imagen bastante clara, vemos que ha detectado muy pocas partes de la imagen.

- **Imagen 7**

Archivo de entrada:



Figura 8-10. Archivo de entrada 7. Fuente: GESTORN

Archivo de salida:

```

NÚMERO MESA

A27 5

A26 5

A25 5

Título del evento

Fecha y hora del evento

Otras informaciones relevantes como por
ejemplo el organizador

14:45:38

```

Figura 8-11. Archivo de salida 7

El formato de la imagen de entrada ha sido jpg. Podemos observar que en este caso ha sido capaz de detectar casi la totalidad de la imagen. Siendo una imagen muy similar a la anterior, difiriendo únicamente en el color del texto, podemos aventurar a afirmar que el software es capaz de detectar mejor cuando existe un mayor contraste.

- **Imagen 8**

Archivo de entrada:



Figura 8-12. Archivo de entrada 8. Fuente: Hospital de l'Esperit Sant

Archivo de salida:

```
H. d'arribada 10:23 FUNDACIÓ HOSPITAL DE
Nom {0} l'Esperit Sant
NOM DEL PACIENT
QM123
La seva cita és a les 0 hores
0
0
consulta 0
1/9 2017 0
167
```

Figura 8-13. Archivo de salida 8

El formato de la imagen de entrada ha sido png. Vemos que, en este caso, del mismo estilo que el anterior, ha sido capaz de identificar casi la totalidad de la imagen.

8.3.1.1 Conclusiones de las pruebas

Una vez vistos los resultados de las diferentes pruebas, saco varias conclusiones. La primera es que no difiere entre el formato de imagen, ya que la misma imagen de entrada con diferentes formatos ha demostrado sacar la misma imagen de salida. Lo que si que incide de manera notable en la detección o no de caracteres es el contraste de estos con el fondo. Hemos visto que imágenes similares con fondo blanco, difiere en la salida dependiendo si el texto se encuentra escrito en color negro o rojo, obteniendo mejores resultados con este último. Con ello, en versiones futuras no sería descabellado

el pasar la imagen por un editor antes de que entre en Tesseract, con el fin de aumentar el contraste de texto con el fondo de manera más notable.

Capítulo 9 - Asignaturas del Máster en Internet de las Cosas

En este capítulo me gustaría hacer un repaso de las asignaturas que he cursado durante el máster y relacionarlas con el presente trabajo.

9.1 Arquitectura del Nodo IoT

Me ha sido de utilidad para el apartado de la red de comunicación y la Raspberry. Además de la conexión de los diferentes dispositivos.

9.2 Diseño de Infraestructura inteligente para el IoT

Gracias al desarrollo multi-plataforma que hemos dado en la asignatura, he conseguido el suficiente empujón para lanzarme a crear el comienzo de una posible aplicación futura.

9.3 Tratamiento de Datos Masivos / Inteligencia Artificial Aplicada al Internet de las Cosas

El reconocimiento óptico de caracteres tiene mucho que ver con estas dos asignaturas, gracias a las cuales he profundizado en una materia de la que he de reconocer que no tenía muchos conocimientos.

9.4 Redes, Protocolos e Interfaces (I & II)

Al igual que con la asignatura vista en el primer punto, el interconexionado de los dispositivos y la red de comunicación, tienen mucho que agradecer a estas dos asignaturas.

9.5 Seguridad y Legalidad

Dado que en el presente trabajo se ha realizado un estudio de una nueva posible utilidad para un futuro, no se le ha otorgado la importancia que merece en el apartado de seguridad.

	ANIOT	DII	TDM / IA	RPI (I, II)	S&L
Red de Comunicación					
OCR					
Aplicación					

Tabla 9-1. Esquema visual relación TFM-Asignaturas

Capítulo 10 - Conclusiones y trabajo futuro

10.1 Introducción

Para concluir con la memoria, se resume la idea general y los conceptos ya descritos en la misma. Con este trabajo se pretende dar una posible solución a un problema real que afecta, contando con las personas que padecen de baja visión, de más de un millón y medio de personas en nuestro país.

10.2 Metas alcanzadas

- **Estudio de la viabilidad de la idea.** Con este trabajo, se ha hecho un análisis de una posible solución, que como hemos visto sería factible de ser usada con éxito en un futuro.
- **Reconocimiento óptico de caracteres.** Se han analizado diferentes opciones y se ha llegado a una solución que presenta un alto porcentaje de acierto, y que además es de código abierto.
- **Raspberry pi.** Se ha probado un "cerebro" del sistema que ha dado buenos resultados para la ejecución del proyecto. Además de realizar pruebas de recepción de imágenes en un dispositivo que carece de esta tecnología.
- **Realización de una pequeña aplicación.** Se ha conseguido desarrollar una parte de la aplicación que el usuario usaría para interactuar con el sistema.

10.3 Problemas encontrados

- Inicialmente, debido a la situación de pandemia que hemos sufrido, no me ha sido posible realizar todas las pruebas que quería del reconocimiento óptico de caracteres en un entorno controlado. Es decir, con un tique al que pudiera haber realizado una fotografía y una pantalla de algún centro medico que pudiera haber fotografiado desde diferentes puntos.

- Continuando con el reconocimiento óptico de caracteres, me hubiese gustado ahondar más en el hecho de que se reconociera el texto de algunas imágenes si, y otras no.
- El desarrollo de la aplicación no ha sido de la forma en que me lo esperaba, en el sentido de que he tardado más de lo esperado en ciertos aspectos.
- La parte de emparejar por bluetooth el móvil a la Raspberry ha de ser de forma automática, estilo la popular red wifi eduroam [49] a nivel universitario, pero usando la tecnología bluetooth. Que el usuario solo tenga que emparejarse una vez y ya le valga para todos los sistemas.

10.4 Trabajo futuro

Para finalizar con el trabajo, se detallan algunas de las cuestiones que se podrían mejorar y otras que se podrían desarrollar más profundamente.

- En primer lugar, la utilización de filtros y máscaras de contorno antes de realizar el reconocimiento óptico de caracteres de la imagen, con lo que probablemente se solventarían muchos de los casos en los que no se detectaba texto.
- Continuación del desarrollo de la aplicación tanto en ANDROID como en IOS.
- Probar con cámaras la transmisión de las imágenes de las pantallas a la Raspberry para su posterior análisis
- En el momento en el que en la Raspberry se tengan las dos imágenes con el texto reconocido, extraer el turno del tique y comparar cada cierto tiempo con el que sale por pantalla.

Capítulo 11 - Introduction (English)

11.1 Motivation

Talking with my thesis's teachers, we reached the chosen topic which I think it is really interesting and could help many people. So, the motivation is double, on the one hand, the fact of get a topic to do this thesis and on the other hand because it is a theme that could be useful for people who really need it.

At first, we had thought about develop a purchase assistant. In detail, design and implement a prototype of supermarket shopping assistant for people with lack of vision. The setting would be the next, supermarket shelf would be marked (with some devices of choice), and the person would receive sound information about the location of the products that wants to get. So, once the person would be in the interest area, the product would be recognized by artificial vision using, for example, the camera of his mobile phone.

Studying this proposal, we realized that the complexity of this project was quite considerable, because all the products would have to be scanned, the camera would have to be placed by the person in the correct way in order to analyze the product etcetera. In addition, the people who suffer this lack of vision probably use online shopping to get their products, so we continue studying the thesis main topic.

11.2 Objectives

Finally, we get the chosen topic, accessible assistant for the management of public spaces. It is a simple idea, any public building that has some type of citizen management which involve making a queue has the typical stand where, once the persona arrives, has to choose the department where needs to go and gets a ticket with a number. Then, enters automatically in a virtual queue. When it is his turn, the number appears on a screen.

Well, what would happen if the person who need to do some procedure in this kind of buildings or go to the doctor (because hospitals are implementing this "virtual queue" thing), had to use this kind of system? For this kind of reasons, we began to

develop this topic. The person would arrive at the building, get the ticket from the stand and would take a picture of it with his mobile phone. The picture would be sent, in this case, to a Raspberry pi 2 model b, where an optical character recognition would get the turn in a language which system could understand. Now we are going to see how the person would know when his turn is and where he should go. In the second part of the design, we have some cameras pointing the screens where turns and rooms appears. We are going to suppose that exist three floors in the building with one possible room per floor. In this assumption we would have also three cameras pointing at their respective screens. These cameras would be connected to our Raspberry. Therefore, continuing with the example, the person has reached the center, has taken a photo of his ticket and that photo has reached the Raspberry. Firstly, after realizing the optical character recognition, would be to know which floor the person has to go. We would get this out thanks to the optical recognition character of the ticket and would be sent to the person application which let him know. Let's think that is the second floor which get out of the recognition, the person would go up to the first floor and the application would tell him that he still has a floor to go up. Once in the third floor, application let him know and that he must wait his turn.

Lastly, we go to the second-floor camera with is focused on turns screen. Camera would take "photos" from time to time or each time positions change and would send them to our Raspberry. At the Raspberry, as with the stands ticket, the images would be entering to the optical character recognition and when screen turns match with tickets turn, it would be notified through the application to user that his turn has come.

This idea is a non-intrusive system, which means could exist a better development in case stands network could be used. But it has been developed in such a way that it does not interfere with what already exists.

11.3 Work plan

Analyzing the prototype in which this work is framed, I have divided it into three branches of study that, from my own point of view, would be the main ones.

To start the communication network, how to connect the devices that come into the system.

Optical character recognition, since it is necessary to obtain the person's turn and compare it with those that appear on the screens.

The application, the way through which the used will interact with the entire system.

In the following table, you can see a Gant chart of how this work has been organized taking into account tasks and weeks.

TASK	W.1	W.2	W.3	W.4	W.5	W.6	W.7	W.8	W.9	W.10	W.11	W.12	W.13	W.14	W.15
Communication Network															
OCR															
Application															
Thesis															

Tabla 11-1. Work organization

11.4 Memory structure

In this section we will see in a general way each of the sections of this document. The structure is the next one:

- **Chapter 1: Introduction.** Introduction of the project in which can find the motivation, with the objective and the work plan.
- **Chapter 2: State of art.** Project and ideas that have something in common with this project are explained.
- **Chapter 3: Resources presentation.** Brief presentation of the devices used for the project tests.
- **Chapter 4: Communication network.** Analysis of different network topologies in order to see which one is the best to this project.
- **Chapter 5: Optical Character Recognition.** Study of different software to do the optical character recognition.
- **Chapter 6: Application.** The developed application test which final used would interact with is presented.
- **Chapter 7: Raspberry Pi.** Analysis of software which runs in this system.
- **Chapter 8. Tests performed.** Realized Tests using optical character recognition program Tesseract.
- **Chapter 9: Master's degree in Internet of Things subjects.** Brief summary which shows what I have learn in the different master's subjects to develop this thesis.
- **Chapter 10: Conclusions and future work.** Conclusions of the present project and future work.
- **Chapter 11: Introduction (English).** Chapter 1 in English.
- **Chapter 12: Conclusions and future work (English).** Chapter 10 in English

Capítulo 12 - Conclusions and future work (English)

12.1 Introduction

To conclude with the thesis, the general idea and the concepts already described in it are summarized. With this project is intended to provide a possible solution to a real problem that affects, counting on people who suffer lack of vision, more than one and a half million people in our country.

12.2 Goals achieved

- **Viability of the idea study.** With this work, an analysis of a possible solution has been made, which, as we have seen, would be feasible to be implemented successfully in the future.
- **Optical character recognition.** Different options have been analyzed and a solution has been reached that has a high percentage of success. Without forget that it is open source.
- **Raspberry pi.** A “brain” of the system has been tested and has given good results for the execution of the project. Receiving images tests has been tested on a device that has not got this technology.
- **Development of a small application.** It has been possible to develop a part of the application that the user would use to interact with the system.

12.3 Problems encountered

- Initially, due to the whole issue of the pandemic that we have suffered, it has not been possible for me to probe all the tests that I wanted to of optical character recognition in a controlled environment. It means, with a ticket and screen from a medical center that could have been photographed by me from different points.

- Continuing with the optical character recognition, I would have liked to delve more into the fact that the text of some images was recognized but others don't.
- The development of the application has not been in the way I expected it. I mean, it has taken longer than expected in certain aspects.

12.4 Future work

To conclude this thesis, some of the issues that could be improved and others that could be developed more deeply are detailed.

- In the first place, the use of filter and contour masks before performing the optical character recognition of the image, which would probably solve many of the cases in which no text was detected.
- To continue with the development of the application in android and IOS system.
- Test the transmission of the images from the screens to the Raspberry with cameras for further analysis.
- When Raspberry system has the two images with re recognized text, extract the turn from the ticket and compare from time to time with the one that appears on the screen.

BIBLIOGRAFÍA

- [1] Raspberry, «Raspberry Pi 2 Model B,» [En línea]. Available: <https://es.rs-online.com/web/p/kits-de-desarrollo-de-procesador-y-microcontrolador/8326274/?sra=pmpn>.
- [2] Wikipedia, «Reconocimiento óptico de caracteres,» [En línea]. Available: https://es.wikipedia.org/wiki/Reconocimiento_óptico_de_caracteres.
- [3] Google, «Empeza a utilizar TalkBack en Android,» [En línea]. Available: https://support.google.com/accessibility/android/answer/6283677?hl=es&ref_topic=3529932.
- [4] C. Collado, «¿Qué es TalkBack en Android y para qué sirve?,» [En línea]. Available: <https://andro4all.com/2019/10/talkback-android-que-es-activar-desactivar>.
- [5] Apple, «VoiceOver,» [En línea]. Available: <https://www.apple.com/es/accessibility/iphone/vision/>.
- [6] Be My Eyes, «Be My Eyes,» [En línea]. Available: <https://www.bemyeyes.com>.
- [7] Microsoft Prensa, «Microsoft presenta Seeing AI,» [En línea]. Available: <https://news.microsoft.com/es-es/2019/12/03/microsoft-presenta-seeing-ai-en-espanol-la-app-gratuita-que-facilita-el-reconocimiento-y-descripcion-del-entorno-a-personas-ciegas/>.
- [8] XATAKA, «Soy ciego y uso la app Seeing IA de Microsoft,» 07 03 2020. [En línea]. Available: <https://www.xataka.com/servicios/soy-ciego-utilizo-app-seeing-ia-microsoft-esto-todo-que-puedo-hacer-ella>.
- [9] Change.org, «Sobre Change.org,» [En línea]. Available: <https://www.change.org/about>.
- [10] I. Calle, «Las nuevas pantallas de los hospitales discriminan a las personas ciegas como yo,» [En línea]. Available: <https://www.change.org/p/las-nuevas-pantallas-de-los-hospitales-discriminan-a-las-personas-ciegas-como-yo-eruzescudero>.
- [11] Wikipedia, «GPIO,» [En línea]. Available: <https://es.wikipedia.org/wiki/GPIO>.
- [12] Raspberry , «Raspberry Pi,» [En línea]. Available: <https://www.raspberrypi.org>.

- [13] cnet, «SMC EX Connect Wireless Bluetooth USB Adapter SMCB-EDR,» [En línea]. Available: <https://www.cnet.com/products/smc-ez-connect-wireless-bluetooth-usb-adapter-smcbt-edr-network-adapter-series/>.
- [14] Google, «Android STUDIO,» [En línea]. Available: <https://developer.android.com/studio>.
- [15] AMAZON, «Just5 Konrow - smartphone 5 pulgadas,» [En línea]. Available: <https://www.amazon.es/Just5-Konrow-smartphone-pulgadas-2-100mAh/dp/B075VCVVMR>.
- [16] Network World, «802.11: estándares de Wi-Fi y velocidades,» [En línea]. Available: <https://www.networkworld.es/wifi/80211-estandares-de-wifi-y-velocidades>.
- [17] C. González, «¿Qué es el WiFi?,» 19 06 2019. [En línea]. Available: <https://www.adslzone.net/reportajes/tecnologia/que-es-wifi-como-funciona/>.
- [18] S. A. Ruiz, «Principales protocolos WiFi y su velocidad máxima,» 06 09 2017. [En línea]. Available: <https://bytelix.com/guias/protocolos-wifi/>.
- [19] Departamento de Cultura Gobierno Vasco, «OCR: Tecnología para el reconocimiento óptico de caracteres en una imagen,» 04 2011. [En línea]. Available: https://www.kultura.ejgv.euskadi.eus/r46-19123/es/contenidos/informacion/kultura2_0_prestakuntza/es_k20_form/adjuntos/pildora-OCR-2.pdf.
- [20] IONOS, «¿Qué es el bluetooth?,» [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-bluetooth/>.
- [21] Yúbal, «Bluetooth: diferencias y características de sus clases y versiones,» [En línea]. Available: <https://www.xataka.com/basics/bluetooth-diferencias-caracteristicas-sus-clases-versiones>.
- [22] Repetidores Móviles, «Matemáticas con dB,» 29 02 2016. [En línea]. Available: <https://blog.repetidoresmoviles.com/matematicas-con-db/>.
- [23] M. J. Gutiérrez, «Todo sobre ZigBee,» 10 08 2015. [En línea]. Available: <https://elandroidelibre.lespanol.com/2015/08/todo-sobre-zigbee-la-tecnologia-ultrabarata-para-comunicacion-inalambrica.html>.

- [24] IngSoftware, «Ventajas y Desventajas de ZigBee,» [En línea]. Available: <https://sites.google.com/site/ingsoftwares5a/ventajas-y-desventajas-de-zigbee>.
- [25] J. Jiménez, «WPAN: Qué es y para qué sirve este estándar de red,» [En línea]. Available: <https://www.redeszone.net/tutoriales/redes-wifi/que-es-estandar-wpan/>.
- [26] D. Cruz, «¿Que es la tecnología Z-WAVE?,» 11 03 2020. [En línea]. Available: <https://tecnosinergia.zendesk.com/hc/es/articles/115002480211--Qué-es-la-tecnolog%C3%ADa-Z-WAVE->.
- [27] DomoDesk, «Z-WAVE sin cables,» [En línea]. Available: <https://www.domodesk.com/162-a-fondo-z-wave-sin-cables.html>.
- [28] Domoticalia, «Z-WAVE,» 07 11 2019. [En línea]. Available: <https://www.domoticalia.es/blog/2019/11/07/z-wave-9-caracteristicas-que-lo-hacen-impresionante/>.
- [29] B. Parthasarathy, «Build your own OCR,» 20 02 2018. [En línea]. Available: <https://medium.com/@balaajip/optical-character-recognition-99aba2dad314>.
- [30] pdfelement, «Qué es el OCR y cómo funciona,» [En línea]. Available: <https://pdf.wondershare.com/es/pdf-knowledge/what-is-ocr.html>.
- [31] A. Sable, «Building Custom Deep Learning Based OCR Models,» 05 08 2020. [En línea]. Available: <https://nanonets.com/blog/attention-ocr-for-text-recognition/>.
- [32] C. G. Moreno, «¿Qué es el Deep Learning y para qué sirve?,» [En línea]. Available: <https://www.indracompany.com/es/blogneo/deep-learning-sirve>.
- [33] F. J. T. N. S.V.Rice, «The Fourth Annual Test of OCR Accuracy, Technical Report 95-03,» Information Science Research Institute, University of Nevada, Las Vegas, July 1995.
- [34] R. Smith, «An Overview of the Tesseract OCR Engine,» Google Inc..
- [35] R. Smith, «A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation,» Proc. of the 3 Int. Conf. on Document Analysis and Recognition (Vol.2), IEEE, 1995.
- [36] A. L. P.J. Rousseeuw, Robust Regression and Outlier Detection, Wiley-IEEE, 2003.

- [37] G. N. T. N. S.V. Rice, Optical Character Recognition: An Illustrated Guide to the Frontier, USA: Kluwer Academic Publishers, 1999.
- [38] P. Schenider, An Algorithm for Automatically Fitting Digitized Curves, Morgan Kaufmann, 1990.
- [39] R. Smith, «The Extraction and Recognition of Text from Multimedia Document Images,» PhD Thesis, University of Bristol, Bristol, November 1987.
- [40] R. Shillman, «Character Recognition Based on Phenomenological Attributes: Theory and Methods,» PhD Thesis, Institute of Technology, Massachusetts, 1974.
- [41] T. K. R. S. B.A. Blesser, «Empirical Test for Feature Selection Based on a Pscychological Theory of Character Recognition,» Elsevier, New York, 1976.
- [42] M. Bokser, «Omnidocument Technologies,» Proc. IEEE, USA, 1992.
- [43] G. Nagy, «At the frontiers of OCR,» Proc. IEEE, USA, 1992.
- [44] Y. X. G. Nagy, «Automatic Prototype Extraction for Adaptive OCR,» Proc of the 4 Int. Conf. on Document Analysis and Recognition IEEE, 1997.
- [45] I. Marosi, «Industrial OCR approaches: architecture, algorithms and adaptation techniques,» Document Recognition and Retrieval XIV, SPIE, 2007.
- [46] The Apache Software Foundation, «Apache Cordova,» [En línea]. Available: <https://cordova.apache.org>.
- [47] Apple, «Developer Apple,» [En línea]. Available: <https://developer.apple.com>.
- [48] Biruntha, «Send Strings From RPI to android device via bluetooth,» [En línea]. Available: <https://www.raspberrypi.org/forums/viewtopic.php?t=131119>.
- [49] eduroam, «eduroam,» [En línea]. Available: <https://www.eduroam.es>.
- [50] Culturacion, «Ventajas y desventajas del WiFi,» [En línea]. Available: <https://culturacion.com/ventajas-y-desventajas-del-wifi/>.
- [51] ScarletApe, «OCR RN,» 24 03 2017. [En línea]. Available: https://github.com/scarletApe/OCR_RN_Tesis.

- [52] H. Míguez, «Reconocimiento de Caracteres con Redes Neuronales,» [En línea]. Available: <https://javamovil.info/post/reconocimiento-de-caracteres-con-redes-neuronales>.
- [53] J. Flesch, «A Python wrapper for Tesseract and Cuneiform,» [En línea]. Available: <https://gitlab.gnome.org/World/OpenPaperwork/pyocr>.
- [54] Firebase, «Reconoce texto en imágenes en Android,» [En línea]. Available: <https://firebase.google.com/docs/ml-kit/android/recognize-text>.
- [55] TIC Portal, «BLOB Binario,» 12 06 2018. [En línea]. Available: <https://www.ticportal.es/glosario-tic/blob-binario>.
- [56] ANDROID, «Cómo tomar fotos,» [En línea]. Available: <https://developer.android.com/training/camera/photobasics?hl=es-419>.
- [57] ANDROID, «Cómo configurar el uso de archivos compartidos,» [En línea]. Available: <https://developer.android.com/training/secure-file-sharing/setup-sharing?hl=es-419>.
- [58] M. V. d. Lucas, «Tesseract-OCR,» 28 12 2013. [En línea]. Available: <https://www.compartolid.es/tesseract-ocr/>.
- [59] Tesseract, «Tesseract OCR,» [En línea]. Available: <https://github.com/tesseract-ocr/tesseract>.
- [60] R. F. H.S. Baird, «A 100-Font Classifier,» Proc. of the 1 int. Conf. on Document aNALYSIS AND rECOGNITION , 1991.
- [61] BytePace, «What is Tesseract and how it works?,» 10 06 2019. [En línea]. Available: <https://medium.com/@Bytepace/what-is-tesseract-and-how-it-works-dfff720f4a32>.
- [62] ANDROID, «Android Basics in Kotlin,» [En línea]. Available: <https://developer.android.com/courses/android-basics-kotlin/course>.
- [63] GESTORN, «Gestor de turnos,» [En línea]. Available: <http://gestorn.com/web/solucion/>.
- [64] I. Calle, «Las nuevas pantallas de los hospitales discriminan a las personas ciegas como yo,» [En línea]. Available: <https://www.change.org/p/las-nuevas-pantallas-de-los-hospitales-discriminan-a-las-personas-ciegas-como-yo-eruzescudero>.

